# Extracting Opinion Targets from User-Generated Discourse with an Application to Recommendation Systems

Vom Fachbereich Informatik
der Technischen Universität Darmstadt
genehmigte

**Dissertation**

zur Erlangung des akademischen Grades Dr.-Ing.

vorgelegt von
**Dipl.-Inform. Niklas Jakob**
geboren in Darmstadt

Tag der Einreichung: 4. April 2011
Tag der Disputation: 18. Mai 2011

Referenten: Prof. Dr. Iryna Gurevych, Darmstadt
Prof. Dr. Gerhard Heyer, Leipzig

Darmstadt 2011
D17

# Ehrenwörtliche Erklärung[1]

Hiermit erkläre ich, die vorgelegte Arbeit zur Erlangung des akademischen Grades "Dr.-Ing." mit dem Titel "Extracting Opinion Targets from User-Generated Discourse with an Application to Recommendation Systems" selbständig und ausschließlich unter Verwendung der angegebenen Hilfsmittel erstellt zu haben. Ich habe bisher noch keinen Promotionsversuch unternommen.

Darmstadt, den 4. April 2011

Dipl.-Inform. Niklas Jakob

[1] Gemäß §9 Abs. 1 der Promotionsordnung der TU Darmstadt

# Wissenschaftlicher Werdegang des Verfassers [2]

| | |
|---|---|
| 10/00–05/07 | Studium der Informatik an der Technischen Universität Darmstadt |
| 12/06–05/07 | Diplomarbeit am Fachgebiet "Telekooperation" an der Technischen Universität Darmstadt "Developing a generic interface for semantic networks" |
| 06/07–01/11 | Wissenschaftlicher Mitarbeiter am Fachgebiet "Telekooperation" und am Fachgebiet "Ubiquitous Knowledge Processing" an der Technischen Universität Darmstadt |

[2] Gemäß §20 Abs. 3 der Promotionsordnung der TU Darmstadt

# Abstract

With the growing popularity of online shopping, most e-commerce websites nowadays offer their customers to leave feedback about their purchases. This form of customer or user interaction is also very popular among Web 2.0 websites. Online databases, e.g. of movies, offer their users incentives to participate in the content creation by giving them the opportunity to rate films and write reviews about them. Complete websites, e.g. rateitall.com, have emerged, which allow their users to rate and review virtually anything they care about. As more and more content is created and aggregated on these websites, a strong demand for automatic approaches which are capable of extracting structured information from mostly unstructured text has emerged. An automatic extraction of the opinions uttered in the thousands of user-generated texts can provide interesting data for several other tasks such as question answering, information retrieval and summarization. All of these tasks require an opinion mining system, which analyzes the individual elements of an opinion on a sentence level, i.e. the terms which express the opinion, their polarity, and what the opinion is about.

In this thesis, we present a comprehensive study of the automatic extraction of opinions with a focus on opinion targets, which is an essential step in order to enable other tasks, e.g. information retrieval or question answering on opinionated content. We analyze the state-of-the-art in opinion mining and divide it into three subtasks, one of which is the extraction of opinion targets. We perform a comparative evaluation of two unsupervised algorithms in the task of opinion target extraction on datasets of customer reviews and blog postings which span the following four different domains: digital cameras, cars, movies and web-services. We show how the identification of opinion expressions influences the opinion target extraction performance of each algorithm. We also show that a simple word distance-based heuristic significantly outperforms both unsupervised algorithms, which make their relevance decision by analyzing word frequencies in the corpus. The word distance-based heuristic reaches an F-Measure between 0.372 and 0.491 on the four datasets.

We furthermore evaluate a state-of-the-art supervised algorithm in the task of opinion target extraction and present a new approach which is based on Conditional Random Fields (CRF). Our approach outperforms the state-of-the-art baseline significantly on all four datasets reaching an F-Measure between 0.497 and 0.702. We also evaluate both algorithms in a cross-domain opinion target extraction task, since a common problem with supervised algorithms is the domain dependence of the learned model. In this setting, our CRF-based approach also outperforms the baseline on all four datasets and it outperforms the best unsupervised approach, which is by design not prone to domain dependence, on three of the four datasets mentioned above. In the cross-domain opinion target extraction task, the CRF-based approach reaches an F-Measure between 0.360 and 0.518 on the four datasets.

The extraction of opinion targets, which are referenced by anaphoric expressions, is a challenge which is frequently encountered in opinion mining at the phrase level. For the first time, we integrate anaphora resolution algorithms in a supervised opinion mining system. We perform a comparative evaluation of two algorithms, in which we require them to extract the correct antecedent of anaphoric targets. Our results indicate that one of the algorithms, which was designed for high-precision

anaphora resolution, is better suited in the opinion mining setting. By extending the algorithm, which yields the best results in the off-the-shelf configuration, we yield significant improvements regarding the extraction of opinion targets on three of the four datasets.[3]

Finally, we show how an opinion mining system can be successfully employed to improve another application. Recommendation systems are nowadays widely used in online platforms and desktop applications in order to suggest goods or pieces of art to users, which they do not know yet, but are likely to enjoy. The recommendations for a user $U_1$ are determined by first profiling the taste and interests of all users of the recommendation system. Then the algorithm identifies other users $U_2 \dots U_n$ which have a similar taste as user $U_1$, and then recommends items to $U_1$ which the users who have a similar taste enjoyed. A user's taste and interests are typically profiled by giving him the option to rate entities, which he has consumed. As mentioned above, website operators have also given users the opportunity to leave their ratings not only on a numerical scale, but also via a free-text review. We hypothesize that these free-text reviews contain a lot of information, expressed in the users' opinions, which would allow us to model his taste and preferences on a very fine granularity. We show that, by integrating our opinion mining system as a feature provider to a state-of-the-art recommendation system, we can significantly improve the accuracy of the recommendations[4], which we evaluate on a dataset of movie ratings and reviews.

[3]Increase of F-Measure between 0.02 and 0.034.

[4]Decrease of root mean square error by ≈1.18%.

# Zusammenfassung

Mit der steigenden Beliebtheit des online Shoppings bieten heutzutage die meisten Betreiber von e-Commerce Webseiten ihren Kunden die Möglichkeit, ein Feedback zu den erworbenen Waren zu hinterlassen. Diese Form der Kunden- oder Benutzerinteraktion ist auf Web 2.0 Seiten stark ausgeprägt. Auf Online-Datenbanken, z.B. für Filme, werden den Nutzern verschiedene Anreize mit dem Ziel geboten, bei der Erstellung der Webseiten-Inhalte mitzumachen. Dabei wird ihnen die Möglichkeit gegeben, Filme zu bewerten und Rezensionen zu schreiben. Es sind mittlerweile Webseiten entstanden, z.B. rateitall.com, die ihren Nutzern ermöglichen Bewertungen und Rezensionen zu den vielfältigsten Themen zu schreiben. Je mehr Inhalte auf derartigen Seiten erstellt werden, desto größer wird der Bedarf an automatischen Ansätzen, die in der Lage sind, strukturierte Informationen aus den meist unstrukturierten Texten zu extrahieren. Eine automatische Extraktion der Meinungen, die in tausenden dieser benutzergenerierten Texten geäußert werden, kann interessante Daten für andere Anwendungen liefern, z.B. Question Answering, Information Retrieval oder automatische Text-Zusammenfassung. All diese Anwendungen erfordern Systeme zur Meinungsextraktion, die in der Lage sind, einzelne Elemente der Meinungen auf Satzebene zu analysieren. Diese beinhalten beispielsweise die Begriffe, welche die Meinung bilden, ihre Polarität und den Betreff der Meinung.

In dieser Dissertation untersuchen wir umfassend die automatische Meinungsextraktion mit einem Schwerpunkt auf der Extraktion von Meinungszielen, da diese ein essentieller Schritt ist, um andere Aufgaben, z.B. Information Retrieval oder Question Answering auf Meinungen durchführen zu können. Wir analysieren den Stand der Forschung im Bereich des Opinion Minings, indem wir die verwandten Arbeiten anhand dreier Teilaufgaben gruppieren. Eine davon ist die Extraktion von Meinungszielen. Wir führen eine vergleichende Studie zwischen zwei unüberwachten Algorithmen zur Extraktion von Meinungszielen durch, die wir auf Datensätzen von benutzergenerierten Rezensionen und Weblog-Postings evaluieren. Diese Datensätze beinhalten Dokumente aus vier verschiedenen Domänen: Digitalkameras, Autos, Filme und Web-Services. Wir analysieren, inwiefern die Identifikation der meinungsbildenden Begriffe die Leistung der Meinungsziel-Extraktion der einzelnen Algorithmen beeinflusst. Des Weiteren zeigen wir, dass eine einfache Heuristik, welche die Wort-Distanz innerhalb eines Satzes zur Identifikation der Meinungsziele verwendet, bessere Ergebnisse als die beiden anderen unüberwachten Algorithmen erzielt. Die Wort-Distanz-basierte Heuristik erreicht dabei ein F-Measure zwischen 0.372 und 0.491 auf den vier Datensätzen.

Ferner evaluieren wir einen Algorithmus, welcher den Stand der Forschung im Bereich der überwachten Meinungsextraktion darstellt. Wir stellen einen neuen überwachten Ansatz zur Meinungsextraktion vor, der auf Conditional Random Fields (CRF) basiert. Der von uns entwickelte Ansatz erzielt auf allen vier Datensätzen eine signifikant bessere Leistung als der Algorithmus nach dem gegenwärtigen Stand der Forschung und erreicht dabei ein F-Measure zwischen 0.497 und 0.702 bei der Extraktion der Meinungsziele. Wir evaluieren weiterhin beide Algorithmen in einem domänenübergreifenden Trainings- / Test-Szenario, da überwachte Algorithmen typischerweise ein inhärentes Problem der Domänenabhängigkeit der gelernten Modelle haben. In diesem Szenario übertrifft unser CRF-basierter Ansatz die Leistung

des Baseline-Systems ebenfalls auf allen vier Datensätzen. Weiterhin vergleichen wir den CRF-basierten Ansatz mit dem besten unüberwachten Algorithmus, der Wort-Distanz-Heuristik, da unüberwachte Ansätze nicht das Problem der Domänenabhängigkeit besitzen. Dabei erzielt der CRF-basierte Ansatz auf drei der vier Datensätze eine bessere Leistung als der unüberwachte Algorithmus. In diesem domänenübergreifenden Szenario erreicht der CRF-basierte Ansatz ein F-Measure zwischen 0.360 und 0.518 auf den vier Datensätzen.

Die Extraktion von Meinungszielen, welche über Anaphern referenziert werden, ist eine Herausforderung, die häufig bei der Meinungsextraktion auf Phrasenebene angetroffen wird. Erstmalig integrieren wir Algorithmen zur Anaphernresolution in ein überwachtes System zur Meinungsextraktion. Wir führen eine Evaluation von zwei Algorithmen durch, bei der diese den korrekten Antezedenten eines anaphorischen Meinungsziels extrahieren müssen. Unsere Ergebnisse zeigen, dass einer der beiden Algorithmen, welcher zur Anaphernresolution mit hoher Präzision entworfen wurde, für eine Integration mit einem System zur Meinungsextraktion geeigneter ist. Indem wir diesen Algorithmus erweitern, der in seiner Standardkonfiguration die beste Leistung erzielt, erreichen wir signifikante Verbesserungen hinsichtlich der Ergebnisse der Extraktion der Meinungsziele auf drei der vier Datensätze.[5]

Abschließend zeigen wir, wie ein System zur Meinungsextraktion erfolgreich verwendet werden kann, um eine andere Anwendung zu verbessern. Empfehlungssysteme werden heutzutage vielfach auf Internet-Plattformen und in Desktop-Applikationen eingesetzt, um den Benutzern Produkte oder Kunstwerke vorzuschlagen, die den Benutzern bisher unbekannt sind, aber gefallen könnten. Die Empfehlungen für einen Benutzer $B_1$ werden berechnet, indem zuerst sein Geschmack bezüglich der Produkte oder Kunstwerke und die Geschmäcker der anderen Benutzer erfasst werden. Der Algorithmus identifiziert dann andere Benutzer $B_2 \dots B_n$, die einen ähnlichen Geschmack wie der Benutzer $B_1$ haben. Dem Benutzer $B_1$ werden dann Entitäten empfohlen, die denjenigen Benutzern gefallen, welche einen ähnlichen Geschmack haben. Der Geschmack eines Benutzers wird typischerweise erfasst, indem man ihm die Möglichkeit gibt, Bewertungen zu den Entitäten abzugeben, die er konsumiert hat. Wie eingangs erwähnt, geht der Trend dahin, den Benutzern die Möglichkeit zu eröffnen, ihre Bewertungen nicht nur auf einer numerischen Skala, sondern auch mittels Freitext auszudrücken. Unsere Hypothese ist, dass diese Freitext-Bewertungen viele Informationen beinhalten, welche in Form von Meinungen ausgedrückt sind, die es uns erlauben, den Geschmack eines Benutzers mit einer sehr feinen Granularität zu modellieren. Wir zeigen, dass sich durch die Integration eines Systems zur Meinungsextraktion die Genauigkeit der Vorschläge eines Empfehlungssystems signifikant verbessern lässt[6]. Dies wurde auf einem Datensatz von Film-Bewertungen und -Reviews evaluiert.

[5] Steigerung des F-Measures zwischen 0.02 und 0.034.

[6] Verringerung der mittleren quadratischen Abweichung um ≈1.18%.

# Acknowledgements

First and foremost, I would like to thank Prof. Dr. Iryna Gurevych and Prof. Dr. Max Mühlhäuser for giving me the opportunity to pursue my doctoral studies. I am especially grateful to Prof. Dr. Iryna Gurevych for her very efficient supervision and her valuable feedback in particular regarding the publications. I would also like to thank all my colleagues at the TK and UKP labs for the nice group spirit, the inspiring talks, discussions and espresso breaks. I want to thank Torsten Zesch and Christof Müller for their work on the ESA implementation, which was used in some of the experiments in this thesis. I am also very grateful to Cigdem Toprak, for her work on the annotation guidelines and corpora collection, Sandra Kübler and the students at the Indiana University for the coordination and realization of the annotation process.

A special recognition needs to be given to the scientific community, which has shared datasets and software - including, but not limited to: Breck Baldwin, Bob Carpenter, Li Zhuang, Feng Jing, Xiao-Yan Zhu, Jason Kessler, Nicolas Nicolov, Bing Liu, Liliana Ferreira and Stefan-Hagen Weber. I would also like to thank everybody involved in the THESEUS Texo project for making it a very fascinating but at the same time smooth project with a pleasant work climate.

I am truly indebted and thankful to my parents for giving me the chance to follow an academic career and their everlasting support and encouragement. It's a great pleasure to thank all my friends and loved ones for everything which happened beyond the IT world. I am obliged to all the artists making the great music which calmed me down or kept me going whichever was necessary.

This work was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ07012".

In all matters of opinion, our adversaries are insane.

Oscar Wilde

Grundsätzlich werde ich versuchen zu erkennen, ob die subjektiv geäußerten Meinungen subjektiv sind oder objektiv. Wenn sie subjektiv sind, dann werde ich an meinen objektiven festhalten. Wenn sie objektiv sind, werde ich überlegen und vielleicht die objektiven subjektiv geäußerten Meinungen der Spieler mit in meine objektiven einfließen lassen.

Erich Ribbeck

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

With the development of Web 2.0 websites and collaborative platforms which encourage users to participate in the generation of content, the structure of the data found on the internet has changed throughout the last years. The utilization of such web communities as an information source has received a lot of attention. This trend was stimulated by the popularity of the integration of customer feedback in online shopping portals or service platforms. Online shopping portals also strongly benefit from the involvement of their customers, since it became very popular for customers to exchange information about purchases in the form of reviews and feedback. In fact, for popular products there can easily be several hundred reviews on larger e-commerce websites as Amazon.com.

Although for the customers it can be more convenient to leave their feedback in a free and unstructured form, this kind of data is most difficult to process by software (Ghose and Ipeirotis, 2007; Ghose et al., 2007). Yet much useful information can be found in e.g. customer reviews, forum discussions or blog postings. One useful type of information available are the *opinions* people express about a given *subject*. These opinions can on the one hand be useful for a potential customer by giving him new insights and allowing a more informed purchase decision, and on the other hand they can be valuable for a vendor since they contain free customer feedback.

While it may be desirable to have available a large number of opinions concerning a given product, the question of managing this data arises. Several approaches for the automatic extraction of information from customer reviews have been presented in the past, many of them focusing on the extraction or summarization of opinions expressed about the product features. The concept *opinion mining* has been coined in the literature (Dave et al., 2003), and it is used to describe the process of automatically extracting opinions from text.

Over the last years, the task of opinion mining has been the topic of many publications. It has been approached with different goals in mind, which require the extraction of opinions on different levels of granularity. In the following, we will divide the related work on opinion mining into three levels of granularity and characterize each of them: Document level, sentence level and word / phrase level.

### Opinion Mining at the Document Level

The coarsest level of granularity comprises an analysis of opinions on a document level. The classification of opinions has been performed on different scales or dimensions:

- **Objective vs. subjective** - classifying e.g. newspaper articles as either containing only factual information vs. opinionated content such as letters to the editor (Wiebe and Wilson, 2002), or identifying opinionated content in an open-domain document collection (Ng et al., 2006; Ni et al., 2007; Godbole et al., 2007)

- **Positive vs. negative** (vs. neutral) - This classification can be either performed as a second step after an objective vs. subjective classification has been done as mentioned above, or e.g. by classifying factual content as neutral (Pang et al., 2002; Titov and McDonald, 2008a; Toprak and Gurevych, 2009).

- **Classifying the document sentiment on a numerical scale** - this task corresponds to a regression problem, in which the overall opinion of a document is to be attributed to e.g. a 5 point scale, which typically ranges from very negative to very positive as frequently encountered in product or movie reviews (Gamon, 2004; Pang and Lee, 2005; Goldberg and Zhu, 2006). This type of classification accounts for a fact that a document can contain a mixture of positive and negative opinions which can lead to an overall average or mediocre impression.

An analysis at the document level was one of the earliest attempts of an automatic opinion analysis and it is still very popular today. This is due to a wealth of labeled data which is available online: Many shopping portals or review websites encourage their users to write free-text reviews and leave an overall numerical rating. This numerical rating is frequently used as a label or representation of the free-text, and a substantial amount of research has investigated its automatic prediction. For algorithmic approaches it can be used e.g. as training data for supervised machine learning systems, but also as a gold standard in the evaluation.

### Opinion Mining at the Sentence Level

The information extracted at this level of granularity is similar to the opinion mining approaches at the document level both in the dimensions of the classification as well as the approaches taken. The related work can be clustered in the following two groups:

- **Objective vs. subjective** classification - Analogue to the classification at the document level, the goal here is to separate factual from opinionated content (Yu and Hatzivassiloglou, 2003; Qu et al., 2008). Consider the following examples:

  (1.1) [Objective] Representatives of many nations attended the G8 summit today.

(1.2) [Subjective] I love how snappy the Safari browser is since the latest update.

- **Positive vs. negative** (vs. both vs. neutral) - Analogue to the classification at the document level, this classification is often performed after the factual content has been separated from the opinions in the previous step (Mullen and Collier, 2004; McDonald et al., 2007). Some work approaches the task as a two-way classification (positive vs. negative), while others consider that a sentence may contain both a positive and a negative statement. Consider the following examples:

  (1.3) [Positive] I love how snappy the Safari browser is since the latest update.

  (1.4) [Negative] Keanu Reeves delivers a disappointing performance in this movie.

  (1.5) [Both] The picture quality of the camera is great but the menu is extremely frustrating to use.

As it is the case for the document level opinion mining, there is a huge body of work on the sentence level classification. Datasets for the evaluation of automatic approaches can be created relatively quickly (Thomas et al., 2006; Seki et al., 2007; Snyder and Barzilay, 2007), and have sometimes even been created by simply using an existing document label and projecting it on its sentences. Notable is the corpus by Pang and Lee (2005), which has been created in that manner and has been employed in numerous related works. The reliability of a gold standard created in this fashion is however questionable, since it e.g. assumes that in a document, which has an overall very positive rating, all sentences also express positive opinions. This can lead to undiscovered misclassifications, if e.g. a sentence which expresses a negative opinion was labeled as positive in the gold standard because it occurs in a document which has been labeled as being positive. A supervised approach might then learn incorrect indicators from that sentence, but in general due to the errors in the gold standard any results obtained on such a dataset can be inconclusive.

### Opinion Mining at the Word / Phrase Level

The task of opinion mining at the word or phrase level includes an identification of the individual elements which form the opinion. This entails that individual words or phrases must be extracted and attributed to an opinion element. The task can be understood as an information extraction (IE) problem (Cowie and Lehnert, 1996), which, on a high level, deals with the extraction of structured information from unstructured text. There are several related tasks or subtasks which can be attributed to the field of information extraction e.g. Named Entity Recognition, Relation Extraction or Terminology Extraction. The task of a Named Entity Recognition system is to identify proper nouns in free text which are then to be attributed to a certain entity category, e.g. person, location, organization (Sang and Meulder, 2003). A Named Entity Recognition (NER) component can be useful for a variety of natural language processing tasks and even directly for other IE tasks such as Relation Extraction: A Relation Extraction system aims at extracting structured

information from free text, which is, in its simplest form, represented in the form of a triple *relation(entity$_1$, entity$_2$)* (Aone and Ramos-Santacruz, 2000; Zelenko et al., 2003). Such relations could e.g. be *causes(gene, disease)* or *acquires(company$_1$, company$_2$)*, and the Relation Extraction system has to discover new instances of them. Obviously a NER component would be employed to identify company names as required in the *acquires(company$_1$, company$_2$)* example.

Another subtask of Information Extraction is Terminology Mining which has the goal of extracting the relevant terms from a given corpus (Bourigault and Jacquemin, 1999; Wermter and Hahn, 2005). "Relevance" is typically defined as specificity in this context, hence with the output of a Terminology Mining system it should be possible to e.g. build a glossary for a corpus or to create a back-of-the-book index.

All of these three IE tasks are related to opinion mining: As it was the case for the Relation Extraction, a NER component can help to identify spans of proper names in text e.g. if an opinion about a person or organization is uttered which we want to extract. Opinion mining can be seen as a variant of a Relation Extraction problem, in which we aim to identify instances of relations such as *likes(person, entity)* for a positive opinion or *dislikes(person, entity)* for a negative opinion. The challenge is however that there is a multitude of ways in which somebody can express his likes or dislikes regarding something (which is commonly referred to as an opinion expression in the opinion mining context). Furthermore, in an opinion mining setting we are typically not only interested in utterances about a certain *entity* as a whole (e.g. a company or a product), but also its aspects / features / parts / facets etc.

The task of opinion mining at the word / phrase level deals with the identification of the following four elements of an opinion, which have been suggested in the related work:

- **Opinion expressions** - These terms or phrases form the opinion and typically express appreciation or dislike of something in a variety of facets regarding strength and differ in polarity. They can emerge from many word classes, e.g. adjectives ("beautiful", "horrible", ...), but also verbs ("love", "hate", ...) and nouns ("masterpiece", "disappointment", ...) (Riloff and Wiebe, 2003; Wilson et al., 2005; Bloom et al., 2007; Breck et al., 2007). In Figure 1.1 the term "likes" expresses the positive opinion which "Paul" has either formulated once or which the author of the sentence believes he holds.

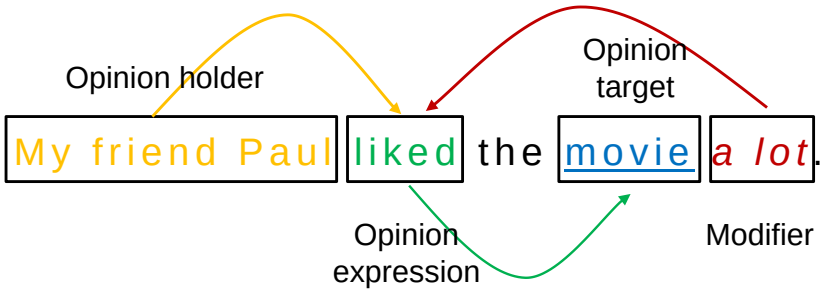
- **Opinion targets** - These are the terms which the opinion of a sentence is about. In reviews these would e.g. be aspects of the entity which is under review or the entity in general (Yi et al., 2003; Hu and Liu, 2004b; Jakob and Gurevych, 2010a). In the example shown in Figure 1.1 "movie" or "the movie" is the target of the opinion. One can imagine that opinion targets can be more abstract concepts such as "the movie", more concrete instances of them e.g. "The Godfather", or again aspects / features / parts / facets of such entities.

- **Relations between opinion expressions and targets** - This step is required in sentences which contain more than one opinion. The goal is then to identify which opinion is about which target, and it is especially important if a sentence contains a mixture of positive and negative opinions (Zhuang

et al., 2006; Kessler and Nicolov, 2009). In the example shown in Figure 1.1, this task is fairly straightforward because there is only one opinion uttered in the sentence, hence we can assume that the opinion expression "likes" refers to the opinion target "movie" (also considering that there are no anaphoric references present in that sentence which could refer to entities from previous sentences).

- **Opinion holders** - The person who utters an opinion is referred to as the opinion holder (Bethard et al., 2004; Kim and Hovy, 2006). In the example from Figure 1.1 "My friend Paul" is the holder of the expressed opinion. The author of this sentence reports someone else's opinion in this example. The identification of opinion holders is mostly pursued in documents which frequently contain reported speech. It is of less importance in reviews which are typically written in order to express the author's point of view.

Figure 1.1: Elements of an Opinion at the Word / Phrase Level




A substantial amount of work has investigated the identification of opinion expressions and subsequently the creation of lexica which contain them. This is due to the fact that even for opinion mining systems which operate on the sentence or document level, an identification of the opinion expressions is an integral step for determining a sentence's or a document's polarity. Lexica of opinion expressions, which define the polarity of a term or phrase, are often employed to achieve this. The automatic creation of such lexica has been investigated in the related work numerous times (Wiebe et al., 1999; Wilson et al., 2005; Esuli and Sebastiani, 2006; Andreevskaia and Bergler, 2006). These lexica attribute words or phrases with a polarity and sometimes strength, which would indicate that e.g. "excellent" expresses a stronger positive opinion than "good". Such lexica are e.g. created by employing manually annotated data, or by bootstrapping from existing (semantic) resources such as WordNet (Fellbaum, 1998) or the General Inquirer (Stone et al., 1966).

In some works, the identification of opinion expressions is also performed in combination with an identification of *opinion modifiers* and *valence shifters*. These are words like "very" or "hardly" which can modify the strength of an opinion expression, as in "very good" or "hardly useful". As it is the case with "hardly" in the previous sentence, these words can also negate the polarity of an opinion expression, so in some cases negation words such as "not", "never" etc. are also treated as modifiers.

The design of an opinion mining system which operates on a phrase level is also dependent of the text domain it will be applied to. Different approaches are taken when mining e.g. newswire in contrast to reviews. Consider the following excerpt from a review of a phone:

(1.6) The salesman in the Verizon store was very friendly and helpful. The good looks of the phone immediately caught my attention.

The difference in the analysis of documents from these domains is that when mining the reviews, one is typically interested in exclusively extracting opinions regarding the entity which is being discussed. The opinion extraction is therefore focused on one product and thereby its attributes and features. The opinion extraction from newspaper documents is different regarding that aspect: Typically all opinions from newswire are being extracted regardless of their target(s).

In this thesis, we will work with user-generated discourse, mainly reviews of products or pieces of art, hence our task is to only extract opinions about the entity under review. We define user-generated discourse as text which was written by an ordinary person on the web and which has not undergone any editorial control. The challenge with user-generated discourse is, that it often contains spelling errors (e.g. improper capitalization, typographical errors etc.), shorthand and abbreviations, slang and web-specific artifacts such as emoticons[1] and grammatical errors.

In Example 1.6, even though the first sentence contains an opinion (about the salesman), we do not want to extract it, as it is not about the entity under review (the phone). Additionally to the identification of the opinions in a given sentence, the algorithms also have to decide whether such an opinion is about the document's overall topic. In this work, we focus on reviews and blog postings which were written in the context of a certain entity (e.g. a movie or a product). It is our goal to only extract opinions which are "on-topic" in the context of a given document (collection) as the one in the second sentence of Example 1.6.

Compared to the vast amount of related work on opinion expression identification, relatively little work has been done on the identification of opinion targets. The task has predominantly been addressed in the context of opinion mining on product or movie reviews. However, the presented algorithms were frequently extrinsically evaluated, due to a lack of annotated data (Hurst and Nigam, 2004; Gamon et al., 2005; Kobayashi et al., 2006; McDonald et al., 2007; Mei et al., 2007; Lu and Zhai, 2008; Titov and McDonald, 2008a). Since several different extrinsic evaluation tasks have been selected in these works, it is difficult if not impossible to infer which of the algorithms actually performs best in the extraction of opinion targets. Since there was only one dataset for the evaluation of opinion mining algorithms on a phrase level available, which only contains documents from one domain, we also developed and conducted an annotation study for the creation of another gold standard, which allows for a comparison of the algorithms' performances across several domains. In this thesis, we will focus on the task of extracting opinion targets and several aspects of it. Hu and Liu (2004b) and Yi et al. (2003) performed an intrinsic evaluation of their opinion target extraction algorithms and we therefore employ their algorithms as baselines in our evaluation.

[1] http://en.wikipedia.org/wiki/Emoticon

## Main Contributions

We now give a brief summary of the main contributions of this thesis:

- We describe the state-of-the-art in unsupervised target extraction and perform a comprehensive analysis of the unsupervised approaches by Hu and Liu (2004b) and Yi et al. (2003), which are based on an analysis of word frequencies in a corpus, for the extraction of opinion targets. In doing so, we investigate how different levels of accuracy regarding the opinion expression identification influence the opinion target extraction. We evaluate the two algorithms on datasets of user-generated discourse which span the following four domains: movies, cars, digital cameras and web-services. We show that, given that the individual opinion expressions have been correctly identified in the sentences, a simple word-distance-based heuristic outperforms the two algorithms, which perform a statistical analysis of word frequencies in corpora, in the task of opinion target extraction.

- The algorithm by Yi et al. (2003) performs an analysis of word frequency ratios between a domain-specific and a general language corpus in order to obtain a relevance ranking for the extraction of the opinion target candidates. We present an extension to this algorithm, which employs an additional corpus extracted from Wikipedia in order to improve the relevance ranking component. We evaluate this extension in three tasks: Product feature extraction, keyphrase extraction and opinion target extraction. We find that our extension yields significant improvements in the product feature extraction and keyphrase extraction tasks, by consistently improving F-Measure across all datasets.

- We propose a new machine learning-based algorithm for opinion target extraction and perform a comparative evaluation of this algorithm against a state-of-the-art supervised system. Our machine learning-based algorithm significantly outperforms the baseline in the task of opinion target extraction. These two algorithms are evaluated on the same datasets as mentioned above, which allows us to also contrast the results of the unsupervised and supervised algorithms. We also investigate the domain dependence of the models learned by the supervised approaches in a cross-domain target extraction scenario.

- We extend a state-of-the-art opinion mining system by an anaphora resolution algorithm. We evaluate the performance of two anaphora resolution algorithms for the extraction of anaphoric opinion targets. We significantly improve the performance of the opinion target extraction by integrating the anaphora resolution component. Furthermore, we present three extensions to one of the anaphora resolution algorithms, which exploit the opinion mining setting and mitigate errors in the antecedent candidate selection. These extensions in turn lead to significant improvements regarding the target extraction performance.

- We integrate an opinion mining algorithm with a recommendation system. We present and evaluate three different algorithms which aim at extracting and

clustering the opinion targets in order to generate features for the recommendation system. These three algorithms require different levels of user interaction, ranging from a fully-manual via semi-automatic to fully-automatic. In order to evaluate the features created by our opinion mining algorithms, we collected a large dataset of movie reviews with corresponding ratings. The integration of our opinion mining-based features yields significant improvements regarding the movie recommendations calculated by the recommendation system.

- We present a scheme and manual for the annotation of opinions on a word / phrase level, which was created in a collaborative research effort. We furthermore collected a dataset of user-generated discourse which was annotated by humans who are not experts in the field of opinion mining, but native speakers of English. Thereby we contributed to the creation of the manual and developed components to crawl the dataset from the web. The corpus annotated in this study exhibits a high inter-annotator agreement and is therefore a reliable resource for an intrinsic evaluation. We also employ this corpus for the evaluation of several algorithmic approaches mentioned above.

## Publication Record

This thesis builds on a number of publications in peer-reviewed conference and workshop proceedings from major events in natural language processing and artificial intelligence, i.e. ACL, EMNLP, ICSC, the WikiAI workshop at IJCAI and the TSA workshop at CIKM.

In (Jakob et al., 2009a), we present an unsupervised algorithm for extracting the relevant terms in a corpus, which we evaluate in the tasks of product feature extraction and keyphrase extraction. In (Jakob and Gurevych, 2010a), we present a machine learning-based algorithm for opinion target extraction and investigate the performance of our algorithm against a state-of-the-art baseline both in a single-domain and cross-domain extraction setting. In (Jakob and Gurevych, 2010b), we investigate the phenomenon of opinion targets which are references by anaphora, and we present a system for the extraction of anaphoric opinion targets. In (Jakob et al., 2009b), we show how an opinion mining system, which is capable of extracting individual components of opinion utterances (e.g. opinion expressions and opinion targets), can be integrated with a recommender system in order to improve its recommendations.

We also actively contributed to the following publications in which the work described in this thesis had an impact on other research: Ferreira et al. (2008) investigate unsupervised approaches for extracting product features from user-generated discourse, which we also employ for the task of opinion target extraction in this thesis. Qu et al. (2008) present a machine learning approach for two tasks: Classifying sentences as objective vs. subjective and in the second step classifying the subjective sentences as being positive / negative / neutral. In Chapters 3 and 4, we investigate how the granularity of the opinion expression identification influences the opinion target extraction. The approach presented in (Qu et al., 2008) could be employed for a sentence level subjectivity classification. We also contributed to the development of the annotation guidelines and the collection of the dataset presented

in (Toprak et al., 2010). The dataset created in this study is employed as a gold standard for evaluation in several chapters of this thesis.

## Thesis Outline

In this thesis, we investigate several aspects of the extraction of opinion targets. In Chapter 2, we describe the datasets of user-generated discourse, which we employ for the evaluation in our opinion target extraction experiments. In Chapter 3, we perform a contrastive evaluation of the two unsupervised algorithms for the extraction of opinion targets by Hu and Liu (2004b) and Yi et al. (2003). We show how different levels of granularity regarding the identification of opinion expressions influence the results of the opinion target extraction. Motivated by the insights gained in our error analysis, we present an extension to the relevance ranking component of the algorithm by Yi et al. (2003), which yields the best results in the opinion target extraction task. We evaluate this extension in two tasks additional to the opinion target extraction, namely product feature extraction - analogue to the work in (Ferreira et al., 2008) - and keyphrase extraction, which is one of the pervasive tasks in natural language processing. We show that our extension significantly improves the results regarding F-Measure in both the product feature and keyphrase extraction task. This suggests, that our extension $\text{LRT}_{wiki}$ performs well in different tasks and for different domains and can be successfully employed if a relevance ranking of words or phrases in a corpus is required.

In Chapter 4, we investigate how supervised algorithms perform in the task of opinion target extraction. We discuss and analyze the algorithm by (Zhuang et al., 2006), which represents the state-of-the-art in supervised opinion target extraction in the movie domain. We then introduce our machine learning-based approach for the extraction of opinion targets, which is based on Conditional Random Fields (CRF) (Lafferty et al., 2001). We perform a contrastive evaluation of these two algorithms and show how our algorithm outperforms the state-of-the-art algorithm on datasets of all four domains. Furthermore, we analyze to what extent the models learned by the supervised approaches are dependent of the domain they were trained on. We achieve this by performing a cross-domain evaluation of the algorithms and again compare the performance with the unsupervised approaches from Chapter 3. We show that our CRF-based approach clearly outperforms the state-of-the-art baseline in the cross-domain setting on all datasets.

In Chapter 5, we integrate the anaphora resolution algorithms by Mitkov (1998) and Baldwin (1997) with a supervised opinion target extraction approach. We show that by integrating and extending the anaphora resolution algorithm by Baldwin (1997), we can reach significant improvements regarding the extraction of opinion targets. Previous opinion target algorithms had not attempted to extract the antecedents referenced by anaphoric opinion targets, but we show that a successful extraction of the targets is possible.

In Chapter 6, we show how the integration of our opinion mining system can improve the results of a recommendation system. We present a dataset of movie ratings with corresponding free-text reviews from which we extract the opinions of the user. These opinions are incorporated into the model learned by the recommendation system as new features, and in doing so, we yield significant improvements regarding

the movie recommendations. We conclude with a summary and some suggestions for future work in Chapter 7. Appendix A describes the annotation guidelines and the annotation scheme which we collaboratively developed in order to create one of the datasets employed for the evaluation of our opinion target extraction experiments.

# Chapter 2

# Datasets

In our experiments in Chapters 3, 4 and 5, we employ datasets from three different sources, which span four domains in total (see Table 2.1). All of them consist of reviews collected from Web 2.0 sites. The first dataset consists of reviews for 20 different movies collected from the Internet Movie Database. It was presented by Zhuang et al. (2006) and annotated regarding opinion target - opinion expression pairs. The second dataset consists of 234 reviews for two different web-services collected from epinions.com, as described by Toprak et al. (2010). The third dataset is an extended version of the data first used by Kessler and Nicolov (2009), which they later describe in more detail in (Kessler et al., 2010). The authors have provided us with additional documents, which have been annotated in the meantime. The version of the dataset used in our experiments consists of 179 blog postings regarding different digital cameras and 336 reviews of different cars.[1] In the description of their annotation guidelines, Kessler and Nicolov (2009) refer to opinion targets as mentions. Mentions are all aspects of the review topic, which can be targets of expressed opinions. However, not only mentions which occur as opinion targets were originally annotated, but also mentions which occur in non-opinion sentences. In our experiments, we only use the mentions which occur as targets of opinion expressions.

All three datasets contain annotations regarding the antecedents of anaphoric opinion targets. In our experimental setups in Chapters 3 and 4, we do not require the algorithms to also correctly resolve the antecedent of an opinion target represented by a pronoun, since in these chapters we are solely interested in evaluating the opinion target extraction without any anaphora resolution. We will address the task of extracting anaphoric opinion targets in Chapter 5.

As shown in rows 4 and 5 of Table 2.1, the documents from the cars and the cameras datasets exhibit a much higher density of opinions per document. 53.5% of the sentences from the cars dataset contain an opinion, and in the cameras dataset even 56.1% of the sentences contain an opinion, while in the movies and the web-services reviews just 22.1% and 22.4% of the sentences contain an opinion.[2] Furthermore, in the cars and the cameras datasets the lexical variability regarding the opinion targets is substantially larger than in the other two datasets: We calculate *target*

[1] The origin of the blog postings and reviews is unknown.

[2] Note that the number of sentences which contain a target is equal to the number of sentences which contain an opinion, because for these two datasets these elements are annotated pairwise.

*types* by counting the number of distinct opinion targets in a dataset. We divide this by the sum of all opinion target instances in the dataset. For a concrete example, assume there is a corpus of reviews in which the users only utter opinions about the "lens cap" and the "viewfinder" of a camera. There are seven opinions which have the "lens cap" as the target and five opinions which are about the "viewfinder". The number of target types in this example is two (since there are two different opinion targets) and the $\frac{target\ types}{targets}$ ratio in this corpus would be $\frac{2}{7+5} = 0.166$. As shown in the last row of Table 2.1 the movies datasets has the lowest lexical variability regarding opinion targets, while the ratio of the web-services dataset is considerably higher with 0.306 and in the cars and the cameras dataset the variability is again higher and the values differ only by 0.007. In terms of reviews, this means that in the movie reviews the same movie aspects are repeatedly commented on, while in the cars and the cameras datasets many different aspects of these entities are discussed, which in turn each occur infrequently.

Table 2.1: Dataset Statistics

| | **movies** | **web-services** | **cars** | **cameras** |
|---|---|---|---|---|
| Documents | 1829 | 234 | 336 | 179 |
| Sentences | 24555 | 6091 | 10969 | 5261 |
| $Average\ \frac{tokens}{sentence}$ | 20.3 | 17.5 | 20.3 | 20.4 |
| Sentences with target(s) | 21.4% | 22.4% | 51.1% | 54.0% |
| Sentences with opinion(s) | 21.4% | 22.4% | 53.5% | 56.1% |
| Targets | 7045 | 1875 | 8451 | 4369 |
| $Average\ \frac{tokens}{target}$ | 1.21 | 1.35 | 1.29 | 1.42 |
| $Average\ \frac{targets}{opinion\ sentence}$ | 1.33 | 1.37 | 1.51 | 1.53 |
| Target types | 865 | 574 | 3722 | 1893 |
| $\frac{target\ types}{targets}$ | 0.122 | 0.306 | 0.440 | 0.433 |

The inter-annotator agreement values reported on the three datasets employed during our experiments can provide some insights on the upper bound of algorithmic approaches. Zhuang et al. (2006) report that the movie reviews were annotated by four "movie fans". An opinion target - opinion expression pair was added to the gold standard if three out of the four annotators agreed on it. The authors report that this was the case in "more than 80% of the sentences". Information regarding annotation guidelines or an annotator training phase is not provided. In Toprak et al. (2010), we describe the annotation process we pursued with our two annotators. The annotation guidelines can be found in Appendix A. In this annotation study, we contributed to the creation of the annotation guidelines and to the data collection. The annotators started with a training phase before the actual annotation process has begun. Each review in the dataset was annotated by the two annotators. For the annotation of opinion targets, the inter-annotator agreement is 0.80 regarding F-Measure, if an exact match of the annotation spans is required. The dataset collected by Kessler and Nicolov (2009) was annotated by four annotators. They

report a less strict agreement by considering overlapping annotations as a match. Consider the following example for the definition of overlapping annotations:

(2.1) The battery life of the Canon G3 is great.

If annotator $A$ selects "battery life" as the opinion target in this sentence and annotator $B$ selects only "battery" or only "life", then this would be counted as a match by the schema of Kessler and Nicolov (2009). The inter-annotator agreement is calculated pairwise for the four annotators. This procedure is performed by taking one annotator as the gold standard and then calculating the overlap with the other one and vice versa. Ultimately, Kessler and Nicolov (2009) report six agreement values, which range between 0.80 and 0.90 regarding F-Measure[3]. Note that the target annotation agreement is only calculated, if the annotators agreed on the opinion expression(s) in the same sentence. This entails, that if there is disagreement regarding the opinion expression(s) in a sentence, then the agreement regarding potential opinion targets will not be calculated. Hence in sentences in which the annotators have disagreement regarding opinion expressions, there cannot be any disagreement regarding opinion targets. The inter-annotator agreement for the opinion expression annotation ranges between 0.72 and 0.85 regarding F-Measure.

A summary of the annotation schemes and inter-annotator agreement values is shown in Table 2.2.

Table 2.2: Datasets: Inter-annotator Agreement

| **Authors** | **Annotation Scheme** | **Inter-annotator Agreement on Opinion Targets** |
|---|---|---|
| Zhuang et al. (2006) | Pairwise annotation of opinion targets and opinion expressions | Not evaluated between individual annotators |
| Toprak et al. (2010) | Sentence- and phrase-level annotation of opinions. Phrase-level consists of several elemens e.g. opinion holder, opinion expression, opinion target, modifier, ... | 0.80 F-Measure between the two annotators, requiring exact match of annotation spans; 0.91 F-Measure requiring only overlapping annotation spans |
| Kessler and Nicolov (2009) | Phrase-level annotation of opinions. Phrase-level consists of several elemens e.g. opinion holder, opinion expression, ... | Between 0.80 and 0.90 F-Measure between the four annotators, requiring only overlapping annotation spans |

[3]Kessler and Nicolov (2009) state that they follow the evaluation metric from (Wiebe et al., 2005) which is essentially the F-Measure because in a comparison of two annotators the recall of $annotator_1$ is the precision of $annotator_2$ and vice versa.

# Chapter 3

# Unsupervised Extraction of Opinion Targets

As mentioned in Chapter 1, the automatic extraction and analysis of opinions has been approached on several levels of granularity throughout the last years. While opinion mining on the document and sentence level is still very popular, some tasks require an extraction and analysis on a term or phrase level. Amongst the tasks which require the finest level of granularity are:

- Question answering - i.e. with questions regarding an entity as in "What does person $P$ like / dislike about $X$?".
- Information retrieval - i.e. returning documents, sentences or individual statements on topic $X$ from the web.
- Summarization - i.e. if one wants to create an overview of all positive / negative opinions from a document collection regarding aspect $Y$ of entity $X$ and cluster them accordingly.

All of these tasks have in common that in order to fulfill them, the opinion mining system must be capable of identifying what the opinions in individual sentences are about, hence extracting the opinion targets.

Our goal in this chapter is to extract opinion targets from user-generated discourse, a discourse type which is quite frequently encountered today, due to the explosive growth of Web 2.0 community websites. Sentences which we encounter in this discourse type are shown in the following examples taken from Zhuang et al. (2006); Toprak et al. (2010); Kessler and Nicolov (2009). The opinion targets which we aim to extract are underlined in the sentences, the corresponding **opinion expressions** are shown in boldface.

(3.1) A long romantic historical extravaganza, and the crowning glory of 1930's movie making, the David Selznick's Gone With the Wind is **worth watching** if only for the color and production design.

(3.2) While none of the features are **earth-shattering**, eCircles does provide a **great** place to keep in touch.

(3.3) Hyundai's **more-than-modest** refresh has largely addressed all the original car's **weaknesses** while maintaining its price **competitiveness**.

## 3.1 Unsupervised Approaches to Opinion Target Extraction

Several approaches for extracting opinion targets, mostly in the context of product reviews have been presented in previous research (Bloom et al., 2007; Carenini et al., 2005; Feiguina and Lapalme, 2007; Holzinger et al., 2006; Kobayashi et al., 2004; Popescu and Etzioni, 2005).

In this chapter, we will focus on unsupervised approaches, which have the advantage of being domain and sometimes even language independent. Although some of the previous research on opinion mining employs an unsupervised algorithm for the actual extraction of opinion targets, many of them rely on external resources. The algorithms by Bloom et al. (2007); Feiguina and Lapalme (2007) and Kobayashi et al. (2004) all rely on pre-built knowledge bases which model the aspects of the e.g. product(s) about which they aim to extract the opinions. As one can imagine, crafting such knowledge bases can be quite time-consuming if done manually and therefore such approaches are not suitable for an open-domain opinion mining system. If the target domain, on which the algorithm shall be employed, must be manually modeled beforehand, then the algorithm cannot be considered as a domain independent approach.

Bloom et al. (2007) manually create taxonomies of opinion targets for two domains. With a handcrafted set of dependency tree paths their algorithm identifies related opinion expressions and targets. Due to the lack of a dataset annotated with opinion expressions and targets for their domain, the authors evaluate the accuracy of several aspects of their algorithm by manually assessing the output on a sample of 200 opinion expressions. Their algorithm yields an accuracy of 0.75 on the sample data in the opinion target identification task. While the accuracy of the opinion target extraction is quite decent, it remains unclear how these values scale up to the entire dataset. Furthermore, the authors do not provide any information regarding their annotation process, hence the conclusiveness of the results is questionable.

Cheng and Xu (2008) also approach the task of identifying product features as opinion targets by using pre-modeled knowledge. They manually create an ontology of concepts for the "car" domain, which is employed as a set of opinion target candidates. The ontology is dynamically enriched during runtime by searching for phrases which match one of their manually defined lexical patterns. They evaluate the performance of their algorithm in a topic extraction scenario, which is similar to a terminology mining task, and independent of the opinion mining task, due to a lack of annotated data. Their algorithm reaches a recall of 0.89 and a precision of 0.94 in the topic extraction task. This suggests that their algorithm performs very well in populating their domain-specific ontology.

Kim and Hovy (2006) aim at extracting opinion holders and opinion targets in newswire with semantic role labeling. They define a mapping of the semantic roles identified with FrameNet (Baker and Sato, 2003) to the respective opinion elements. As a baseline, they implement an approach based on a dependency parser,

which identifies the targets following the dependencies of opinion expressions. They measure the overlap between two human annotators and their algorithm as well as the baseline system. The algorithm based on semantic role labeling yields an F-Measure of 0.315 with annotator1 and 0.127 with annotator2, while the baseline yields an F-Measure of 0.107 and 0.109 regarding opinion target extraction. The relatively low F-Measure of their algorithm is due to a very low recall of 0.2 and 0.07, however the precision is considerably higher with 0.64 and 0.58. The authors are motivated by these results and they conclude that a more sophisticated analysis of the relationship between opinion expressions and opinion targets is necessary.

But there are also approaches which do not rely on manually crafted knowledge: The algorithm by Popescu and Etzioni (2005) (OPINE) calculates the probability that a candidate term is relevant in a certain domain using a statistical analysis. The domain (e.g. "digital camera") is represented by a term or phrase and they employ Point-wise Mutual Information as a measure of relevance. The relevance score is calculated by performing an analysis of term co-occurrence statistics on a large corpus, typically the web. In order to calculate the co-occurrence statistics, a web search engine which supports the *NEAR* operator (proximity search) in queries is required. Popescu and Etzioni (2005) combine this information with the output of a non public web-scale information extraction (IE) system. In their evaluation, Popescu and Etzioni (2005) benchmark OPINE against the algorithm by Hu and Liu (2004a) (see Subsection 3.1.1), which they outperform by +0.22 regarding precision reaching a value of 0.99. But OPINE's recall is 0.03 lower reaching a value of 0.77. However, the web search engine they employed in their experiments does not support the proximity search any more. This drawback in combination with the private IE system makes the great results of Popescu and Etzioni (2005) very difficult if not impossible to verify and also not very flexible regarding deployment.

Feiguina and Lapalme (2007) work on the extraction of product features as opinion targets from customer reviews. They present an algorithm based on an information extraction system, which is neither dependent of search engine nor the web as a corpus. Their information extraction system learns a language model on part-of-speech patterns. This is achieved by training it on a dataset which has been labeled regarding e.g. entities and their aspects. The information extraction system will then learn the part-of-speech pattern which connects the *entity* and the *aspect*, e.g. for the phrase "the viewfinder of the camera" it would learn "NN IN DT NN", given that "viewfinder" is labeled as an aspect and "camera" is the entity. They evaluate their algorithm in a cross-domain setting, however only precision values of the terminology extraction are reported, which is independent of an opinion mining step, hence the performance of the opinion target extraction is unknown.

Titov and McDonald (2008a) present two extensions to the LDA algorithm by Blei et al. (2003), which is a generative approach to topic modeling. The assumption of LDA is, that each document in a corpus consists of a mixture of an arbitrary number of topics, and that each word is attributable to one of these topics. The user has to specify the number of topics which the LDA algorithm shall extract, and the algorithm will determine the topics by clustering the words in the corpus around them. Their first extension (MG-LDA) is described in (Titov and McDonald, 2008b), which is capable of modeling two distinct types of topics: global topics and local topics. The intention behind this extension is, that the algorithm can be

applied to a corpus of e.g. reviews which do not only discuss different entities of one domain (e.g. digital cameras), but can instead be applied to a corpus which contains reviews from different domains (e.g. cars, digital cameras and movies). Their extension MG-LDA shall then be able to 1.) identify the different domains in the corpus and 2.) identify and attribute individual aspects of the entities discussed in a given domain. These entity aspects can then serve as target candidates in an opinion mining setting. The second extension presented in (Titov and McDonald, 2008a) builds upon MG-LDA, and is capable of identifying entity aspects and extracting them as opinion targets, as well as performing a sentiment analysis step. In both (Titov and McDonald, 2008a) and (Titov and McDonald, 2008b) the authors can however only perform an extrinsic evaluation of their algorithms, since the datasets they work with are not annotated regarding individual opinion expressions or opinion targets.

Table 3.1 provides a summarizing overview of the discussed approaches.

Table 3.1: Comparison of Unsupervised Approaches for Opinion Target Extraction

| **Author** | **Core Method** | **Evaluation & Results** |
|---|---|---|
| Bloom et al. (2007) | Handcrafted domain taxonomy + dependency tree paths | intrinsic, accuracy 0.75 on output sample of 200 opinion expressions on customer reviews |
| Cheng and Xu (2008) | Handcrafted ontology of domain concepts as target candidates + ontology population during runtime | extrinsic, precision 0.94 recall 0.89 in topic extraction task on customer reviews |
| Kim and Hovy (2006) | Semantic role labeling | intrinsic, F-Measure 0.315 with annotator1 and F-Measure 0.127 with annotator2 on newswire |
| Popescu and Etzioni (2005) | Information extraction system + web search engine | intrinsic, precision 0.99 recall 0.77 on customer reviews |
| Feiguina and Lapalme (2007) | Information extraction system | extrinsic, precision 0.80 in terminology extraction task on customer reviews |

While we will analyze both supervised and unsupervised approaches for opinion target extraction in this thesis, in this chapter we will focus on unsupervised methods. These have the advantage that they do not require any manually labeled training data and do not depend on any hand-crafted domain specific knowledge. Therefore, they are generally applicable to data from any domain. We are interested in approaches which do not make any assumptions regarding a certain target domain and if possible have little or no language specific requirements since we have datasets from multiple domains available which we can perform a comparative evaluation on. To our knowledge, there exist two domain and language independent approaches that do not rely on hand-crafted or world knowledge. We will employ them in our experiments and hence elaborate on them in more detail in the following sections.

### 3.1.1 Association Mining

One of the earliest works on opinion target extraction was done on customer reviews of consumer electronics. Hu and Liu (2004a) introduce the task of *feature based summarization*, which aims at creating an overview of the product features commented on in the reviews. Their approach relies on a statistical analysis of the review terms based on association mining (Agrawal and Srikant, 1994). This algorithm was originally designed for so called "shopping cart" or "market basket analysis", which aims at identifying interesting combinations of items which are frequently bought together in a supermarket. In such a study, an american drug store once discovered that on weekdays in the early evening typically diapers and beer are bought together. An unexpected combination, which was attributed to the shopping habits of young fathers.

The system of Hu and Liu (2004a) uses association mining to identify the feature candidates of the products occurring in product reviews. In analogy to the shopping cart analysis, the words of a sentence represent the bought items. The algorithm then mines correlations regarding their occurrences. Hu and Liu (2004a) assume that the product features occur as nouns and that the opinions about these features are expressed by adjectives. A distinction is made between so called *frequent features* and *infrequent features (iff). Frequent features* appear in several documents, while *infrequent features* are commented on less often. In the following, we will elaborate on how these two feature types are identified.

**Identifying Frequent Features**

Association mining (Agrawal and Srikant, 1994) is employed in order to extract the frequent features. The association mining algorithm calculates the probability that certain features or feature sets occur in the review document collection for a certain product. Candidate terms for both kinds of features are nouns only. The nouns occurring in a sentence are used to create a so called *transaction set*, which corresponds to the items bought by a customer in the shopping cart analysis. The transaction sets from all reviews of a certain product are used as input for the association mining algorithm. So in the first step the algorithm will create $t_1 \dots t_i$ transaction sets where $i$ is the number of sentences in the dataset. A given transaction set $t_a$ will contain all nouns (=feature candidates) from sentence $a$. For a

dataset of camera reviews, the transaction sets could e.g. look like the example shown in Table 3.2.

Table 3.2: Transaction Set Examples

| **Transaction Set #** | **Elements** |
|---|---|
| $t_1$ | camera, lens, shop |
| $t_2$ | viewfinder, lens, flash |
| $t_3$ | camera bag, flash, picture quality |
| $t_4$ | camera, flash, picture, noise |
| $t_5$ | salesman, display, window |
| $\vdots$ | $\vdots$ |
| $t_i$ | wife, camera, baby, picture |

The algorithm then cycles through all transaction sets and counts the total number of feature candidates. From this total, the empirically defined threshold of 1% is calculated which is employed for the *minimum support*. Then for each feature candidate $fc$ in all transaction sets $t$ the total number of occurrences is calculated. For the abovementioned example these occurrences could be as follows:

Table 3.3: Element Occurrence Counts

| **Element** | **Occurrence Count** |
|---|---|
| camera | 88 |
| lens | 52 |
| shop | 12 |
| flash | 75 |
| camera bag | 13 |
| picture | 77 |
| picture quality | 27 |
| noise | 9 |
| salesman | 2 |
| display | 33 |
| window | 3 |
| wife | 12 |
| baby | 8 |
| $\vdots$ | $\vdots$ |

Assuming that there are 926 feature candidates in total, the minimum support would then be 9.26. The algorithm then goes through all feature candidates and extracts those with a number of occurrences greater than 9.26, so in the example above everything but “noise”, “salesman”, “window” and “baby”. These features

which have a number of occurrences greater than the threshold are extracted as *frequent features*.

Since association mining does not consider the position of the terms in sentences, two pruning steps are applied: The first pruning step is called *compactness pruning*. It removes *frequent feature sets* in which the individual terms do not occur within a distance of three or less words in two or more sentences of the document collection. So in the example above, the algorithm would go though all sentences and check whether for "camera bag" and "picture quality" the individual words "camera" and "bag" / "picture" and "quality" occur within a distance of three or less words in two or more sentences of the document collection. The goal of this pruning step is to remove noun sets which do not occur as a noun phrase in at last two sentences. The second pruning step, called *redundancy pruning*, removes frequent features or frequent feature sets which are complete subsets of other *ffs*, if the subset does not occur by itself in three or more sentences. So in the example above the feature "picture" is a candidate for the redundancy pruning because it is contained in "picture quality". However since it occurs 77 times, it will not be removed. The goal of this pruning step is to only keep longer noun phrases, which expected to be more concrete entities, e.g. "battery life" is more meaningful than only "life" in a corpus of digital camera reviews.

**Identifying Infrequent Features**

*Infrequent features* are extracted from the sentences which do not contain any frequent features, but contain an opinion expression. For identifying opinions about the product features, Hu and Liu (2004a) follow a lexicon-based approach. Based on the previous work on the correlation of subjectivity and the presence of adjectives in sentences (Bruce and Wiebe, 1999; Wiebe et al., 1999), opinion words are assumed to be adjectives. The lexicon of opinion words is created by crawling WordNet (Fellbaum, 1998) starting from an empirically defined set of seed adjectives[1]. By crawling synonyms and antonyms of the seed adjectives in WordNet, a list with 99 positively and 111 negatively oriented adjectives is created. In sentences which contain an opinion word, but no frequent feature, the noun(s) with the smallest distance (in words) to the opinion word is/are extracted.

### 3.1.2 Likelihood Ratio Test

Yi et al. (2003) present the *Sentiment Analyzer* algorithm which identifies product features by extracting a set of base noun phrases as candidate feature terms and ranks them according to a relevance score. The Likelihood Ratio Test (LRT) was introduced by Dunning (1993) and has been employed for many different NLP-related tasks. Yi et al. (2003) employ it in order to calculate the relevance of a given product feature and ultimately to extract opinions about it. The algorithm does not assume that the population it operates on is distributed normally or approximately normally, which is true for the frequencies of terms in a text. The Likelihood Ratio Test identifies relevant terms from a document collection by comparing the frequencies of the candidate terms in the "on-topic" documents with their frequencies in a

[1]The authors state that they bootstrap from 30 "very common adjectives", a concrete list is not provided

general language "off-topic" document collection. The algorithm creates a contingency table for the current candidate term $T$. This table contains $C_{11}$ which is the term's frequency in the "on-topic" document collection $D_+$ and also $C_{21}$ which is the term's frequency in the "off-topic" document collection $D_-$. $C_{12}$ and $C_{22}$ are the sums of the frequencies of all other terms in the respective document collections. Table 3.4 visualizes the different elements of the contingency table.

Table 3.4: Contingency Table for Candidate Term $T$

| | $\mathbf{D_+}$ | $\mathbf{D_-}$ |
|---|---|---|
| $\mathbf{T}$ | $C_{11}$ | $C_{12}$ |
| $\overline{\mathbf{T}}$ | $C_{21}$ | $C_{22}$ |

Using these values, the LRT is defined as follows:

$$
\begin{aligned}
-2\log\lambda &= \begin{cases} -2 * lr & \text{if } r_2 < r_1 \\ 0 & \text{if } r_2 \geq r_1 \end{cases} \\
r_1 &= \frac{C_{11}}{C_{11} + C_{12}} \\
r_2 &= \frac{C_{21}}{C_{21} + C_{22}} \\
r &= \frac{C_{11} + C_{21}}{C_{11} + C_{12} + C_{21} + C_{22}} \\
lr &= (C_{11} + C_{21})\log(r) + (C_{12} + C_{22})\log(1 - r) - C_{11}\log(r_1) \\
&\quad - C_{12}\log(1 - r_1) - C_{21}\log(r_2) - C_{22}\log(1 - r_2)
\end{aligned}
\tag{3.4}
$$

As shown in Equation (3.4), the quotients $r$, $r_1$ and $r_2$ consider the sizes of the document collections $D_+$ and $D_-$, by dividing the occurrences of the candidate term by the sum of all term occurrences. The higher the value of $-2\log\lambda$, the higher the likelihood that the $T$ is relevant in the corpus $D_+$. The LRT can be used to calculate the relevance of individual words or phrases. Yi et al. (2003) empirically define a set of part-of-speech sequences which are used to extract the candidate words and phrases. The relevance for these candidates is then calculated using the LRT.

### 3.1.3 Comparison of the Approaches

Table 3.5 presents a comparison of the two approaches discussed above, summarizing the individual steps of each of them. We observe that the Association Mining approach is less restrictive in the selection and extraction of candidate features. The part-of-speech patterns restrict candidate terms for multiword features to consecutively occurring nouns, while the Association Mining approach can combine nouns occurring anywhere in a sentence to a multiword feature. This characteristic of the association mining creates more flexibility compared to the Likelihood Ratio Test approach concerning multiword feature extraction, but at the same time introduces

Table 3.5: Comparison of Approaches for Product Feature Extraction

| | **Likelihood Ratio Test** | **Association Mining** |
|---|---|---|
| Candidate selection | Patterns of POS sequences | Nouns |
| Candidate ranking | Based on Likelihood Ratio Score | No, only *minimum support* threshold |
| Depends on opinion identification | No | Partly |
| Uses empirically defined threshold | Yes, for extraction threshold | Yes, for *minimum support* |
| Considers position of feature in a sentence | Yes | Partly with compactness pruning |
| Can extract multiword features | Yes | Yes |
| Requires general vocabulary corpus | Yes | No |

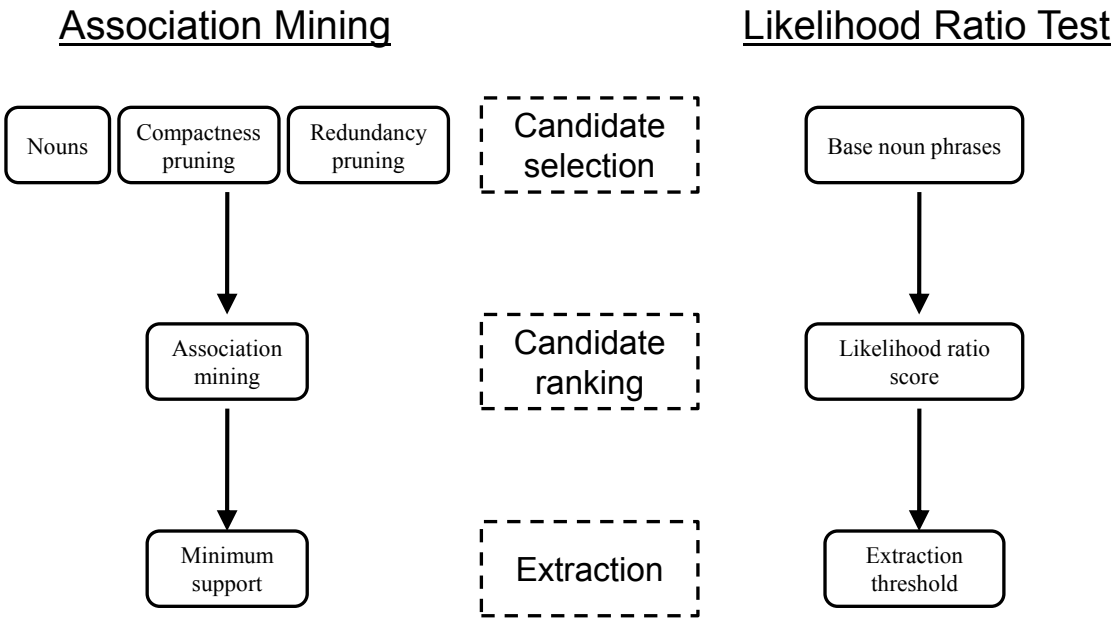
a new source of potential errors. Therefore the employment of the compactness pruning step is necessary[2]. Both approaches rely on a threshold which affects the feature selection, for which it is not possible to calculate an ideal value in advance. Figure 3.1 outlines an overview of the components used for candidate selection, ranking and extraction of the two approaches.

## 3.2 Comparative Study of Association Mining and Likelihood Ratio Test

In the following, we will evaluate the algorithms by Hu and Liu (2004a) and Yi et al. (2003) regarding their opinion target extraction performance. Hu and Liu (2004a) originally evaluated their entire system on a manually selected subset of the dataset of product reviews which they annotated. They do not state which documents or sentences they selected for the evaluation, hence we cannot reproduce their results. Furthermore, their opinion target extraction is dependent of their component for opinion expression extraction, namely for the identification of *infrequent features*. The results of Hu and Liu (2004a) indicate that identification of infrequent features considerably improves the recall of the entire system. However, they cannot evaluate the performance of their opinion expression component, since the opinion expressions are not identified and annotated in their dataset. The algorithm by Yi et al. (2003)

[2]In the experiments of Hu and Liu (2004a), the compactness pruning yielded an increase of precision by 0.10 while decreasing recall by only 0.01.

Figure 3.1: Architecture of Target Extraction Approaches




does not rely on the identification of opinion expressions for the extraction of opinion targets. As mentioned above, they do not report the recall of their algorithm in the evaluation, hence an important factor of the performance remains unclear. In the following, we will evaluate the two algorithms in the task of opinion target extraction. The individual opinion expressions are annotated in all datasets which we employ in our experiments, hence we can evaluate the Association Mining algorithm by Hu and Liu (2004a) independent of the actual approach used for the opinion expression identification. We also evaluate the Likelihood Ratio Test based system by Yi et al. (2003) in the task of opinion target extraction. Our goal is to perform a comparative evaluation of these two unsupervised algorithms, and we strive to present more conclusive results than previous research.

### 3.2.1 Experimental Setup and Metrics

Both algorithms make certain assumptions about the word classes of the opinion target candidates (see Sections 3.1.1 and 3.1.2). We employed the Stanford POS-Tagger (Toutanova et al., 2003) for both tokenization and part-of-speech tagging. The input for both algorithms was lowercased and lemmatized, for which employed the TreeTagger (Schmid, 1994) with the default model[3].

We employ the following requirements in our evaluation of the opinion target extraction: An opinion target must be extracted with exactly the span boundaries

[3]We found that the Stanford POS-Tagger employs a more sophisticated tokenization component than the TreeTagger, therefore we utilized it instead of performing all tasks with the TreeTagger.

as annotated in the gold standard. This is especially important regarding multiword targets. Consider the following example in which the opinion target is "auxiliary input jack":

(3.5) The radio features a very useful auxiliary input jack.

If an algorithm only extracts e.g. "jack" or "auxiliary" as the opinion target, the meaning of the statement is lost, but evaluation strategies which allow partial matching would treat this as a correctly extracted target. Therefore in our evaluation we require that the complete phrase is extracted as it is annotated in the respective datasets. Extracted targets which partially overlap with the annotated gold standard are counted as errors. We did not make any exceptions e.g. for articles since there are cases for example movie titles ("The Two Towers") where the article definitely belongs to the entity (or not). Hence a target extracted by the algorithm, which does not exactly match the boundaries of a target in the gold standard, is counted as a false positive (FP). Referring back to Example 3.5, if only "auxiliary" or "jack" are extracted as targets, both will be counted as FPs. Exact matches between the targets extracted by the algorithm and the gold standard are true positives (TP). We refer to the number of annotated targets in the gold standard as $T_{GS}$. Precision is calculated as $Precision = \frac{TP}{TP+FP}$, and recall is calculated as $Recall = \frac{TP}{T_{GS}}$. The F-Measure is the harmonic mean of precision and recall.

**Setup for the Likelihood Ratio Test Approach:**

As a collection of topical documents ($D_+$), we employ the respective documents in the dataset on which we are currently evaluating the algorithm, as described in Table 2.1. As non-topical documents ($D_-$), approximately 600 documents were randomly selected from the UKWaC British English web corpus (Ferraresi et al., 2008). Yi et al. (2003) define three types of noun phrases which they employ to identify target candidates: Base noun phrases (BNP) which are a combination of the following part-of-speech tags[4] as defined by the Penn Treebank (Marcus et al., 1993): NN | NN NN | JJ NN | NN NN NN | JJ NN NN | JJ JJ NN. Definite base noun phrases (dBNP) are base noun phrases which are preceded by the definite article "the" in a sentence. Beginning definite base noun phrases (bBNP) are definitive base noun phrases which occur at the beginning of a sentence. The algorithm by Yi et al. (2003) only returns a set of target candidates with the corresponding likelihood ratio scores in the first step. Hence a threshold for the selection of target candidates is required. The target candidates, which have a likelihood ratio score higher than the threshold, shall then be extracted subsequently. Yi et al. (2003) define $n$ as the number of dBNPs in the given dataset and then use the $n$ BNPs with the highest likelihood ratio scores. In (Jakob et al., 2009a), we have shown that this threshold selection approach can result in a high precision extraction, but typically leads to a low recall. Therefore, we proposed and evaluated a different approach for threshold selection, which is based on the algorithm for outlier detection presented in (Wilcox, 2001, page 38). Our approach for the threshold calculation outperforms

[4]Pattern sequences are separated by the pipe symbol |

the approach taken by Yi et al. (2003), because it analyzes the actual distribution of the likelihood ratio scores. The threshold $t_{LRT}$ is set to:

$$t_{LRT} = m_{lr} + sd_{lr} \tag{3.6}$$

where $m_{lr}$ is the mean likelihood ratio score and $sd_{lr}$ is the standard deviation of all BNPs in the current dataset.

**Setup for the Association Mining Approach:**

Since the association mining disregards the original order of the terms in sentences, we cannot reconstruct whether the extracted *frequent feature set* [picture, quality] occurred as "quality picture" or "picture quality" in the dataset. For the evaluation, we therefore match every permutation of an extracted *ffs* against a multiword target in the annotation. If one term order results in a match, we count that as a correct result, otherwise it is considered a false result. If the returned feature is just a subset or subsequence of the annotated feature we consider that a false result too. Hu and Liu (2004a) do not report how they evaluate multiword targets. We employ the same values for the minimum support and the compactness pruning as suggested in (Hu and Liu, 2004a).

### 3.2.2 Results

In the following, we will present the results of the algorithms in the task of opinion target extraction. We evaluate the algorithms in four different settings:

I: Extraction of opinion targets when opinion bearing sentences have been identified with perfect accuracy.

II: Extraction of opinion targets when individual opinion expressions in sentences have been identified with perfect accuracy.

III: Extraction of opinion targets without any information about opinion expressions.

IV: Extraction of opinion targets when individual opinion expressions in sentences have been identified with a domain-independent state-of-the-art subjectivity lexicon.

With these four settings, we aim to evaluate opinion target extraction algorithms in both synthetic and real-world settings. As synthetic settings we consider Settings II and III. In these two settings, we can evaluate the opinion target extraction performance independent of the (usually foregoing) identification of opinion expressions. From the results of Settings II we can gain insights regarding the upper bound of the opinion target extraction of each algorithm. In Setting III on the other hand we investigate a "worst-case" scenario, given that the opinion target extraction has to be performed without any information regarding the opinion expressions available. This setting is similar to other information extraction tasks such as keyword extraction or Named Entity Recognition. With the Settings I and IV,

we aim to evaluate the performance of the algorithms in a real-world scenario. The related work on sentence-level opinion mining has shown that the identification of opinion bearing sentences is possible with a high accuracy across different domains[5]. In Setting I, we will investigate the upper bound of the algorithms while emulating that e.g. a classifier has been run beforehand for the identification of opinion bearing sentences. With Setting IV, we aim to evaluate the algorithms in a scenario which lies between Settings II and III. Although a purely lexicon-based approach does not represent the state-of-the-art in opinion expression identification, we rely on it in this setting. In doing so, we expect to gain some insights regarding the influence of the foregoing opinion expression identification on the opinion target extraction performance of the different algorithms.

Both the Association Mining based approach and the Likelihood Ratio Test based approach return a list of candidate terms / phrases which are selected as relevant in the dataset under analysis and are extracted as opinion targets. In Setting I, we will extract the candidate terms / phrases only in sentences which contain at least one opinion expression. For the identification of *infrequent features*, the Association Mining based approach requires that individual opinion expressions have been identified in the given dataset. Hence in Setting I, we can only evaluate the *frequent feature* identification of the Association Mining based approach. As presented in (Yi et al., 2003), the Likelihood Ratio Test based approach is independent of the opinion expression identification and can therefore fully be evaluated in Setting I.

In Settings II and IV, individual opinion expressions in the sentences are available. Hence we will evaluate the complete Association Mining based approach as presented in (Hu and Liu, 2004a).

**Results Setting I**

Table 3.6: Target Extraction with Gold Standard Opinion Sentences

| | **A**ssociation Mining (*frequent features*) | | | **L**ikelihood Ratio Test | | |
|---|---|---|---|---|---|---|
| **Dataset** | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| movies | 0.411 | 0.353 | 0.380 | 0.325 | 0.721 | 0.448 |
| web-services | 0.274 | 0.259 | 0.266 | 0.293 | 0.476 | 0.362 |
| cars | 0.179 | 0.112 | 0.138 | 0.207 | 0.355 | 0.262 |
| cameras | 0.194 | 0.189 | 0.191 | 0.233 | 0.392 | 0.292 |

The results of the opinion target extraction of the two algorithms are shown in Table 3.6. As only the sentences which contain an opinion expression are identified in this setting, we can solely employ the *frequent feature identification* component of the Association Mining based algorithm. The Likelihood Ratio Test based approach is employed as introduced by Yi et al. (2003). We observe that both algorithms yield

[5] Yu and Hatzivassiloglou (2003); Qu et al. (2008) reach an F-Measure of ∼0.85 in the identification of opinion bearing sentences in blog postings and newswire.

the best results regarding F-Measure on the dataset of movie reviews, followed by the dataset of web-services. The Likelihood Ratio Test based approach outperforms the Association Mining based approach on all four datasets regarding F-Measure. This is due to a consistently higher recall of the Likelihood Ratio Test based approach. As outlined in Section 3.1.1, a threshold of 1% regarding the minimum support is set for the extraction of *frequent features* in the Association Mining based approach. While this threshold may seem rather low, depending on the distribution of the opinion targets quite a few of the targets might not be added to the *frequent feature* list. Especially on the "cars" and the "cameras" datasets, the recall of the Association Mining based approach is very low. As discussed in Chapter 2, the $\frac{target\ types}{targets}$ ratio is especially high in these datasets, which suggests that many targets will only occur very seldomly. The more dynamic calculation of the extraction threshold, based on the distribution of the likelihood ratio scores, we employ for the Likelihood Ratio Test yields better results in this setting. The precision of both algorithms is surprisingly low on both datasets, given that in this setting, the information whether a sentence contains an opinion is taken from the gold standard annotation. The list of target candidates has between 30 and 40 entries depending on the algorithm and dataset. Given that there are between 574 and 3722 target types in the four datasets, this list is very compact. The candidates also occur in many opinion sentences in which they are not the actual targets, hence the algorithms extract them as false positives. Consider the following example:

(3.7) I highly recommend this site to anyone looking for a way to keep in touch with friends or family.

Now assume that both "site" and "friends" are identified as opinion target candidates. The algorithms will extract both terms from this sentence, although only "site" is the target of the opinion expression ("highly recommend").

### Results Setting II

Table 3.7: Target Extraction with Gold Standard Opinion Expressions

| | Association Mining (*frequent + infrequent features*) | | | Likelihood Ratio Test + nearest noun phrase | | |
|---|---|---|---|---|---|---|
| **Dataset** | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| movies | 0.425 | 0.465 | 0.444 | 0.324 | 0.727 | 0.449 |
| web-services | 0.298 | 0.339 | 0.317 | 0.293 | 0.499 | 0.369 |
| cars | 0.258 | 0.272 | 0.265 | 0.209 | 0.376 | 0.269 |
| cameras | 0.233 | 0.279 | 0.254 | 0.237 | 0.416 | 0.302 |

Table 3.7 shows the results of the target extraction with the opinion expressions taken from the gold standard annotation. In this Setting, we can also employ the *infrequent feature* identification step of the Association Mining based approach. We observe that the *infrequent feature* identification improves both the precision and

the recall of the target extraction on all four datasets. This was not the case in the evaluation by Hu and Liu (2004a), where the inclusion of the *infrequent feature* identification consistently resulted in a loss of precision. However, the results of the configuration with only the *frequent feature* identification were much higher regarding precision in (Hu and Liu, 2004a).

We can now compare the results of the Association Mining based approach as presented in (Hu and Liu, 2004a) with the results of the Likelihood Ratio Test based approach as presented in (Yi et al., 2003) from Table 3.6. We observe that the results of the Association Mining based approach with both the *frequent* and *infrequent feature* identification are still consistently lower regarding F-Measure. The Association Mining based approach yields higher results regarding precision on three datasets, but the results regarding recall are still considerably lower. The inclusion of the *infrequent feature* identification step cannot fully compensate for the low recall of the *frequent feature* identification which we already observed in the results of Setting I.

What Hu and Liu (2004a) introduce as their *infrequent feature* identification is basically a heuristic which attempts to extract the opinion target based on distance to the opinion expression in the sentence, if the Association Mining did not identify a target candidate for a sentence beforehand. This step is independent of the Association Mining, and it is hence possible to also combine it with other (unsupervised) target extraction approaches.

Since it consistently improves both precision and recall of the Association Mining approach, we combine this heuristic with the Likelihood Ratio Test based approach as follows: If the Likelihood Ratio Test based approach does not identify a target candidate in an opinion sentence, we extract the noun phrase which is closest (regarding distance in words) to the opinion expression as the target. The results of this approach are shown in the right three columns of Table 3.7. We observe that the results slightly improve regarding F-Measure when compared to the results of the original Likelihood Ratio Test. The improvements are however not statistically significant[6]. We conclude that the Likelihood Ratio Test based approach can yield better results than the Association Mining based approach, while being less dependent of the opinion expression identification and thus being more robust.

### Results Setting III

Since there is no information about any opinion expressions available in this Setting, we can only employ the *frequent feature* identification step of the Association Mining based approach as it was the case in Setting I. As shown in Table 3.8, the performance of both algorithms regarding precision considerably declines on all datasets. This shows, that the target candidates which both algorithms select also occur without any opinion being mentioned about them quite frequently. The results reflect the ratio of opinion sentences in the respective datasets: The decline of the target extraction precision is less on the "cars" and "cameras" datasets. We attribute this

[6]Significance of improvements is tested using a paired two-tailed t-test with $p \leq 0.05$. In most tables in this Chapter, we cannot denote the statistical significance of improvements in the default *-notation directly on the results since we often compare several configurations. We will instead report the statistical significance of any improvements in the result discussions.

Table 3.8: Target Extraction without any Opinion Identification

| | Association Mining (*frequent features*) | | | Likelihood Ratio Test | | |
|---|---|---|---|---|---|---|
| **Dataset** | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| movies | 0.115 | 0.353 | 0.173 | 0.085 | 0.721 | 0.152 |
| web-services | 0.068 | 0.264 | 0.108 | 0.071 | 0.488 | 0.124 |
| cars | 0.104 | 0.113 | 0.108 | 0.119 | 0.359 | 0.179 |
| cameras | 0.133 | 0.190 | 0.157 | 0.162 | 0.398 | 0.231 |

to the higher density of opinion sentences in these two datasets. The overall highest results regarding precision and F-Measure are reached on the "cameras" dataset. This shows, that in this dataset, the highly ranked target candidates are quite frequently actual opinion targets and do seldomly occur while not being the target of an opinion.

**Results Setting IV**

Table 3.9: Target Extraction with MPQA Opinion Expressions

| | Association Mining (*frequent features*) | | | Likelihood Ratio Test | | |
|---|---|---|---|---|---|---|
| **Dataset** | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| movies | 0.126 | 0.310 | 0.179 | 0.093 | 0.631 | 0.162 |
| web-services | 0.089 | 0.171 | 0.117 | 0.091 | 0.317 | 0.141 |
| cars | 0.157 | 0.071 | 0.098 | 0.179 | 0.224 | 0.199 |
| cameras | 0.155 | 0.120 | 0.135 | 0.192 | 0.262 | 0.221 |
| | Association Mining (*frequent + infrequent features*) | | | Likelihood Ratio Test + nearest noun phrase | | |
| movies | 0.113 | 0.401 | 0.176 | 0.092 | 0.634 | 0.160 |
| web-services | 0.089 | 0.214 | 0.126 | 0.089 | 0.327 | 0.140 |
| cars | 0.187 | 0.140 | 0.160 | 0.176 | 0.234 | 0.201 |
| cameras | 0.166 | 0.161 | 0.163 | 0.190 | 0.275 | 0.225 |

In this Setting, our goal is to investigate to which extent a correct identification of opinion expressions influences the target extraction. As we have shown in the results of Setting II, for the Likelihood Ratio Test based approach an identification of individual opinion expressions does not yield any improvements regarding the target extraction, while for the Association Mining based approach an identification

of individual opinion expressions is necessary so that it reaches the performance of the Likelihood Ratio Test based approach. We will now not rely on the gold standard annotations, but instead employ the freely available MPQA lexicon (Wilson et al., 2005) for the identification of opinion expressions. With over 8000 entries the coverage of the lexicon is quite substantial, and it has been successfully employed in several prior works (Eguchi and Lavrenko, 2006; Kim and Hovy, 2006; Qu et al., 2008; Mihalcea et al., 2007; Popescu and Etzioni, 2005; Choi et al., 2006; Kanayama and Nasukawa, 2006; Chesley et al., 2006). The lexicon distinguishes between "strong" and "weak" subjectivity clues. Previous research (Qu et al., 2008; Toprak and Gurevych, 2009) has shown that the strong subjectivity clues are good indicators for opinion expressions across several domains. The subset of strong subjectivity clues still contains 5569 entries, which distinguishes between different part-of-speech forms and also covers conjugations and declensions of the entries. These 5569 entries consist of 4747 types, as some entries occur in different part-of-speech forms. The distribution of the part-of-speech forms is as follows: 3.93% adverbs, 15.46% verbs, 25.85% nouns, 36.02% adjectives and 18.72% of the entries are tagged as "anypos" which means, that they are considered to be subjectivity clues regardless of in which part-of-speech form they occur. Table 3.10 shows some random sample entries from each part-of-speech class.

Table 3.10: Sample Entries from the MPQA Lexicon

| **Part-of-speech** | **Subjectivity Clue Examples** |
|---|---|
| adverbs | blissfully, comfortably, erratically, flatteringly, inexcusably |
| verbs | impeach, irritate, maximize, praise, stupify |
| nouns | abhorrence, avalanche, fallacy, pacifist, pain |
| adjectives | abusive, impatient, painstaking, trustworthy, spiritless |
| anypos | enthralled, hatefully, heroically, stench, valuable |

In analogy to our analysis in Settings I and II, we will first evaluate the performance of the opinion target extraction when only information about the opinion sentences is available. Then, in the second setting we will employ the individual opinion expressions. We consider sentences which contain one or more opinion expressions as identified by the MPQA lexicon as opinion sentences.[7]

The upper four result rows of Table 3.9 show the performance of both algorithms with the opinion sentences being identified with the MPQA lexicon. We observe that analogous to the results of Setting III (Table 3.8), the *frequent feature* identification of the Association Mining based algorithm slightly outperforms the Likelihood Ratio Test based approach on the movies dataset regarding F-Measure, while yielding slightly lower F-Measure scores on the other three datasets. Furthermore, we observe that even in the best configurations in Table 3.9, there are hardly any improvements regarding the precision when the MPQA lexicon is employed, compared to the setting without any opinion identification shown in Table 3.8. At the same

[7]An analysis of this assumption and the performance of the MPQA lexicon for opinion expression identification follows in Section 3.2.3.

Table 3.11: Target Extraction with Nearest Noun Phrase Heuristic

| | Opinion Expression Identification | | | | | |
|---|---|---|---|---|---|---|
| | Gold Standard | | | MPQA | | |
| **Dataset** | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| movies | 0.517 | 0.469 | 0.491 | 0.105 | 0.386 | 0.164 |
| web-services | 0.420 | 0.335 | 0.372 | 0.102 | 0.190 | 0.133 |
| cars | 0.387 | 0.396 | 0.391 | 0.231 | 0.156 | 0.186 |
| cameras | 0.433 | 0.438 | 0.436 | 0.236 | 0.179 | 0.204 |

time, the MPQA lexicon does probably not cover all opinion expressions of the four corpora and hence leads to a loss of recall regarding the target extraction. We will investigate this in the error analysis of this chapter. The coverage is especially bad for the "cameras" dataset, on which the loss of recall outweighs the gain of precision, which results in a lower F-Measure than in Setting III, in which no information regarding the opinions is available.

The lower four result rows of Table 3.9 show the performance of the Association Mining based approach in the complete configuration and the Likelihood Ratio Test based approach extended by the nearest noun phrase heuristic. We observe that the inclusion of the nearest noun phrase heuristic yields a decrease of F-Measure for both algorithms on the "movies" and "web-services" datasets. On the "cars" and "cameras" datasets, we observe a consistent increase of F-Measure for both algorithms, however for the Likelihood Ratio Test based approach the improvements are not significant[8]. In this Setting, the Association Mining based approach still outperforms the Likelihood Ratio Test based approach regarding F-Measure on the "movies" dataset. However, on the other three datasets the Likelihood Ratio Test based approach yields higher F-Measure scores. In Setting III, the Likelihood Ratio Test based approach yields the best results regarding F-Measure on the "cameras" dataset with 0.231. If we compare this result to the best performing configuration in Table 3.9, we observe that on this dataset the loss of recall introduced by the MPQA lexicon even outweighs the gain in precision, leading to an overall lower F-Measure score. The effects which errors in the opinion expression identification introduce are quite striking on this dataset. The loss of recall is to be expected, as it is unlikely that the MPQA lexicon covers all opinion expressions from any domain. However the gain of precision which we would expect from using opinion expressions for the extraction of opinion targets is very small compared to the results of Table 3.8.

### Results Nearest Noun Phrase Heuristic

Motivated by the consistent increase of F-Measure which the *infrequent feature* identification / *nearest noun phrase* component introduced in the results of Setting II,

[8]Significance of improvements is tested using a paired two-tailed t-test with $p \leq 0.05$.

we will now investigate how this heuristic performs in isolation. In this additional setting, we will hence not employ any of the algorithms, but extract opinion targets with only the *nearest noun phrase* heuristic. The nearest noun phrase heuristic is formally defined in Algorithm 1.

**Algorithm 1** Nearest Noun Phrase Heuristic

```
for all OpinionExpressions in currSentence do
  nearestNounPhrase = null
  minDistance = ∞
  for all NounPhrases in currSentence do
    currDist = countWordDistance(currOpinionExpr, currNounPhrase)
    if currDist < minDistance then
      nearestNounPhrase = currNounPhrase
      minDistance = currDist
    end if
  end for
  if nearestNounPhrase ≠ null then
    label nearestNounPhrase as opinion target for currOpinionExpr
  end if
end for
```

As this approach requires an identification of individual opinion expressions we will, analogue to the results above, evaluate it with the gold standard opinion expressions and the opinion expressions as identified by the MPQA lexicon. Table 3.11 shows the results of these experiments. As shown in the three leftmost result columns, the performance of this approach is quite competitive regarding F-Measure if the gold standard opinion expressions are employed. If we compare these results to the best performing configuration of the experiments above (Likelihood Ratio Test + *nearest noun phrase*) in Table 3.7, we observe that the *nearest noun phrase* heuristic in isolation outperforms the Likelihood Ratio Test. This is due to a higher precision in the opinion target extraction. Both the Association Mining based approach and the Likelihood Ratio Test based approach generate many false positives because highly ranked target candidates quite frequently occur in opinion sentences in which they are not the actual targets as shown in Example 3.7.

The results of the opinion target extraction with the *nearest noun phrase* heuristic based on the MPQA lexicon employed as a resource are shown in the three rightmost result columns of Table 3.11. Compared to the configuration in which we employ the gold standard opinion expressions we observe a considerable loss in precision on all datasets. The recall also strongly decreases on the "web-services", "cars" and "cameras" datasets. Thus compared to the configuration with the gold standard opinion expressions, we observe very low F-Measure scores. If we compare the results with the best performing configurations of Setting IV (Table 3.9), we observe that the results of the *nearest noun phrase* heuristic in isolation come relatively close to the statistical approaches. This is due to slightly higher precision scores of the *nearest noun phrase* heuristic. Again, we attribute this to the false positives introduced by the statistical approaches due to highly ranked target can-

didates which occur in opinion sentences, but are not the actual opinion targets as shown in Example 3.7.

### 3.2.3 Error Analysis

We will present our error analysis split into four parts. First, we will discuss some typical sources of errors which affect all approaches and can decrease both precision and recall. We will then analyze the sources of errors which mostly have a negative impact on the recall of the target extraction in Section 3.2.3 and on the precision in Section 3.2.3. In Section 3.2.3, we will investigate the cause for the low results during target extraction when the MPQA lexicon is employed for opinion expression identification. Finally, we will analyze the errors which occur during the target extraction with the *nearest noun phrase* heuristic in Section 3.2.3.

#### False Negatives in Target Extraction

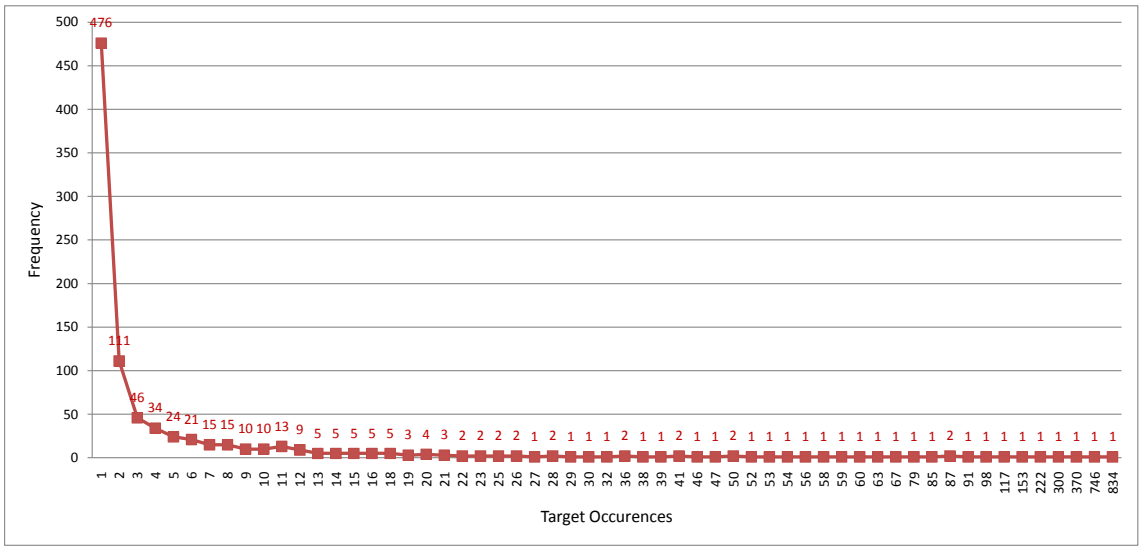
As a first step we want to analyze why, even in the best configuration, the recall of the two unsupervised approaches for target extraction is so low, especially on the "web-services", "cars" and "cameras" datasets. Both algorithms employ the number of occurrences of an opinion target candidate as a factor to rank its relevance. The Association Mining based approach even employs a fixed threshold, which a target candidate must reach regarding occurrences, in order to be considered during the *frequent feature identification* step. In order to analyze possible problems of the algorithms we created histograms of the opinion target type distributions for each dataset, shown in Figures 3.2, 3.3, 3.4 and 3.5. All graphs follow the same layout: The opinion target types are grouped by their number of occurrences $o_n$ in the respective dataset on the x-axis. The y-axis shows the number of opinion targets types which occur $o_n$ times in the corpus. The $y$ value of a data point is shown above it. As an example, in Figure 3.2 the leftmost data point shows, that there are 476 different opinion targets which occur once in the dataset. These are e.g. terms such as "technical achievement", "dramaturgy", individual actor's names e.g. "Michael C. Hall", but also misspellings e.g. "dialoge" in the movie domain. The rightmost data point shows that there is one opinion target in the dataset which occurs 834 times. In this domain, these are very frequent terms such as "movie", "film", "characters" or "actors". Note that the values on the x-axis are typically not linear, but instead there are gaps.

In general, we observe that the opinion target types exhibit a Zipfian distribution on all datasets. In the following, we will discuss them individually.

As shown in Figure 3.2, in the "movies" dataset there are 476 opinion targets which only occur once and 111 target types which occur twice. These $476 * 1 + 111 * 2 = 698$ targets account for 9.9% of the overall 7045 targets. Assuming that these rare target types do not occur very often while not being opinion targets in the corpus, both algorithms presented above would rank them very low or not even add them to the candidate list.

As mentioned in Section 3.1.1, the Association Mining based approach employs a threshold of 1%, as the minimum support for the *frequent feature* identification. 1% equals an occurrence threshold of 70 in this dataset. As shown in the histogram, only

Figure 3.2: Histogram of Target Distribution in "movies" Dataset




the 12 rightmost datapoints reach the threshold, while there are two target types which occur 87 times, which means that there are only 13 opinion target types added to the *frequent feature* set. All of this assuming, that there are no other nouns / noun phrases which occur that often in the dataset. On the other hand, these 13 opinion target types account for 3296 targets in the dataset, which equals 46.8% of the overall targets. This value indicates the upper bound regarding recall for the Association Mining based approach on the "movies" dataset when only the *frequent feature* identification is employed. Judging from the results in Table 3.6, there are quite a few opinion target types which the algorithm could theoretically extract, but does not, since it only reaches a recall of 0.353. One cause could be that the approach by Hu and Liu (2004a) employs individual nouns as input candidates to the Association Mining. If the Association Mining (re)combines multiword targets in the wrong order, or not all orders in which they occur in the text, then the algorithm will subsequently not extract them. Furthermore, the *redundancy pruning* step will discard target candidates which are included in other, longer, multiword targets, e.g. "life" if "battery life". However, as shown in the histogram in Figure 3.2, there are many ($476 + 2 * 111 + 3 * 46 = 836$) targets which only occur less than four times, which is the threshold for the *redundancy pruning*. Hence, if one of these rare targets also occurs in another multiword target candidate, it will not be extracted and generate a false negative.

For the Likelihood Ratio Test, a prediction regarding possible target candidates is less straightforward, since it employs the general language corpus $D_+$ in order to perform the relevance ranking of a target candidate. Therefore, in order to estimate the rank of a given target candidate we would need to make assumptions regarding its frequency in the off-topic document collection $D_-$ which is not possible. We can however safely assume that target candidates which e.g. occur five times or less in

the on-topic document collection $D_+$ cannot yield a high enough Likelihood Ratio Score in order to be extracted, even if they do not occur at all in $D_-$. As mentioned above, the threshold for the target candidate set is calculated using the average of the Likelihood Ratio Scores and its standard deviation. As shown in Figure 3.2, there are also some target types which occur very often hence typically have a very high Likelihood Ratio Score (unless they are also terms / phrases from the general vocabulary). These very frequent target candidates will therefore also raise the average Likelihood Ratio Score. For example in the "movies" dataset the most frequent targets "film" and "movie" have a Likelihood Ratio Score of over 11000, while the average score is approximately 10.42. In order to reach this threshold, a phrase would have to occur at least six times in $D_+$, while not occurring in $D_-$ at all. Naturally a frequency count $> 0$ in $D_-$ raises the bar even higher.

Figure 3.3: Histogram of Target Distribution in "web-services" Dataset

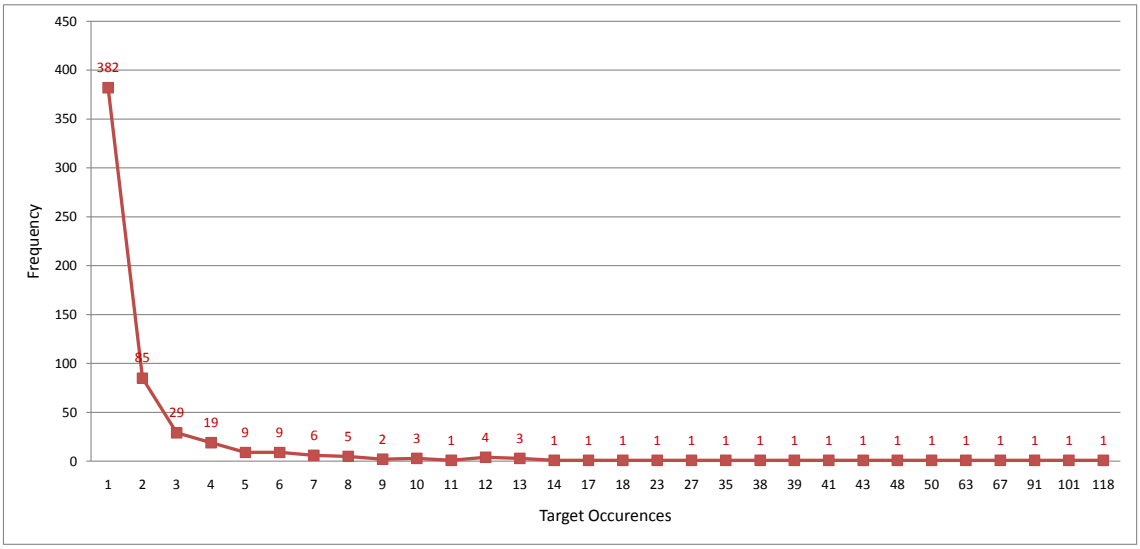
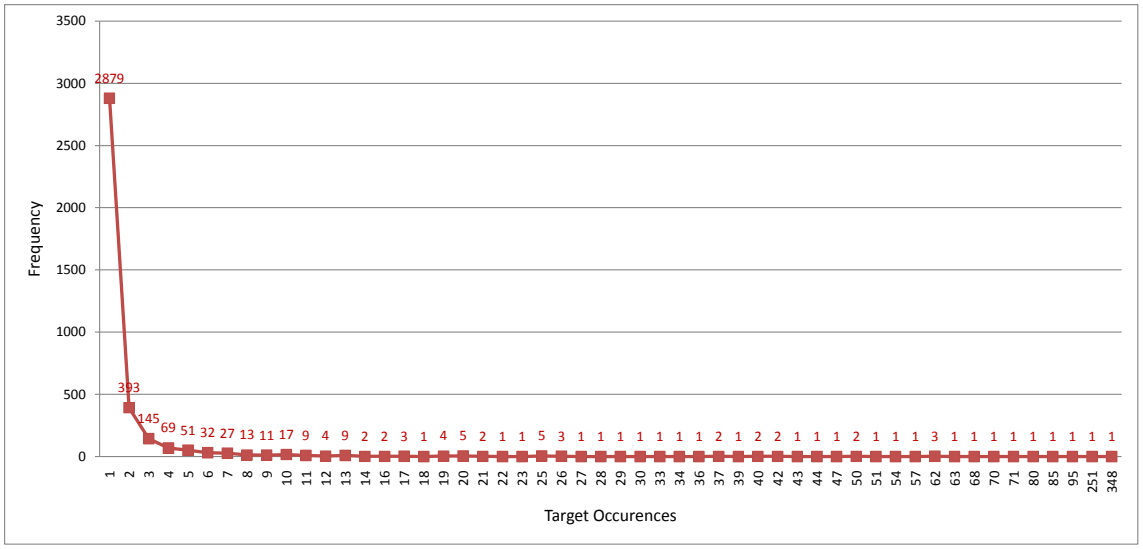



Figure 3.3 shows the histogram of target type distributions for the "web-services" dataset. We observe a pattern which is similar to the histogram of the "movies" dataset. The total number of target types which occur only once is lower, but since the dataset is also a lot smaller, these target types account for 20.4% of the overall targets already. Again, both algorithms cannot extract those targets without the *infrequent feature identification / nearest noun phrase* heuristic. For the Association Mining, the *minimum support* threshold is 19 on this dataset and due to its characteristics described above, the Likelihood Ratio Test will never rank target candidates which only occur once high. As shown in Table 3.6, the recall of the best performing configuration is already considerably lower than on the "movies" dataset. The average Likelihood Ratio Score in this dataset is 13.81, which requires at least four occurrences in $D_+$. If we analyze the share of target types which reach this threshold analogous to above by calculating $\sum_{n=4}^{n=118} frequency_n * occurrences_n$, we observe that with a perfect candidate selection and ideal distributions in $D_-$ the

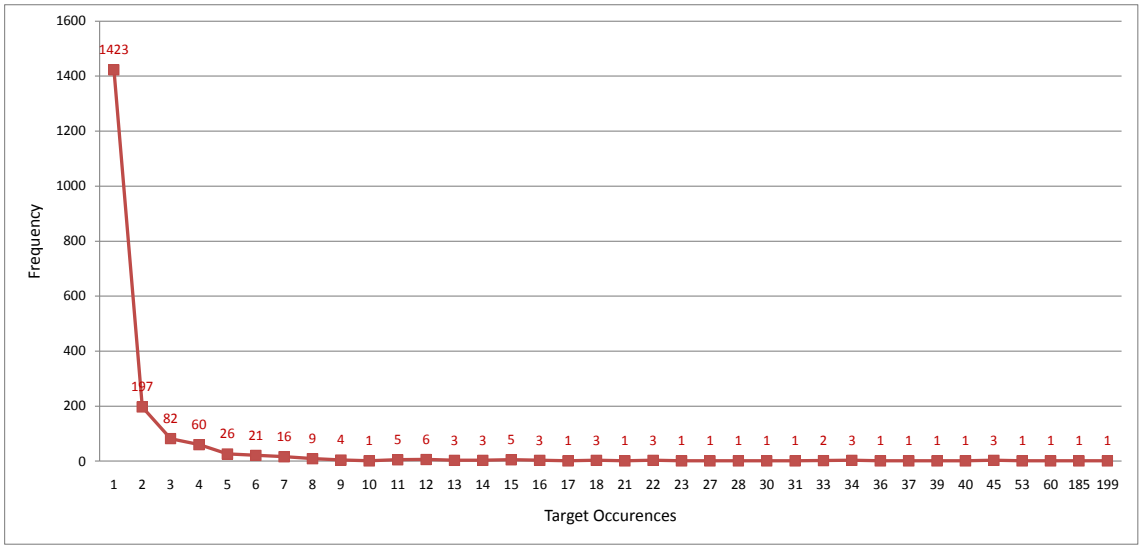
Likelihood Ratio Test based approach displays an upper bound of 61.8% regarding recall for the target extraction.

Figure 3.4: Histogram of Target Distribution in “cars” Dataset




When analyzing the target histogram of the “cars” dataset in Figure 3.4, we observe that the target types which occur only once already account for a large number of the overall targets. If we calculate the sum of the target types which occur up to five times, we observe that those account for 54.8% of all targets. An algorithm which fails to extract these rare targets has hence an upper bound of 0.452 regarding recall assuming that its candidate selection and ranking is perfect.

Figure 3.5: Histogram of Target Distribution in “cameras” Dataset



As shown in Figure 3.5, the distribution of the target type frequencies in the "cameras" dataset is very similar to the "cars" dataset. This is also reflected by our experiments, in which the recall values of the best performing configurations are typically very similar. The sum of the frequencies of the target types which occur five times or less account for 55.6% of all targets. This indicates why the recall of the two algorithms is so low. A summary of the statistics regarding the rare opinion targets is compiled in Table 3.12.

Table 3.12: Share of Targets Occurring Five Times or Less

| **Dataset** | **Share of Rare Targets** |
|---|---|
| movies | 15.50% |
| web-services | 40.53% |
| cars | 54.79% |
| cameras | 55.68% |

We believe that the problem which both algorithms have with the extraction of rare opinion targets is very challenging. A possible solution might be an algorithm which also considers rareness as an indicator of relevance such as $\chi^2$. However it is questionable how such an algorithm performs on user-generated discourse, in which we have to expect quite a few spelling errors etc., which are then prone to be extracted by the algorithm as well. Another solution might be to refer to another source of information for the extraction of rare targets. Ideally, this would be an existing knowledge base, which should on the one hand cover many domains and on the other hand be of high textual quality in order to avoid the problem with misspellings. Such a knowledge base could be employed as an "authority" to verify whether a rare target candidate is a misspelling or not. If a certain rare target candidate can also be found in the knowledge base, then the probability that it is a misspelling is low, given that the knowledge base exhibits a high textual quality.

### False Positives in Target Extraction

As shown in Tables 3.6 and 3.7, the precision of both algorithms is very low, even if the information regarding opinion sentences and expressions is taken from the gold standard. We first analyzed whether this is due to false entries in the algorithm's target candidate lists. A manual inspection of the target candidate lists has shown, that they do not or hardly contain any false positives. An analysis of a sample of sentences which contain false positives has shown that the problem is more complex: Both algorithms typically select terms which are very frequent in a given dataset as target candidates, e.g. "camera" or "shot" in the "cameras" domain. We have observed that these terms are so frequent that they occur in almost every sentence, and in many cases they are not the target of the opinion. Some examples are shown in the following sentences, in which the actual targets are underlined and false positives are in italics:

(3.8) So far, I'm quite satisfied with this *camera*'s performance.

(3.9) In outdoor *shots*, colours are very good and natural.

In our error analysis, we encountered many sentences such as Example 3.8, in which people comment on a certain attribute of the entity under discussion, thereby mentioning the general entity (in this case "camera") as well. This is also typically achieved by constructs as "The performance of the camera is [...]". A detection of these attribute - entity relations could either be conducted by employing an additional knowledge base which models this type of information or by defining a set of grammatical or phrase rules which aim to identify them. At the same time there are of course many cases in which "camera" is actually the opinion target in a sentence as in:

(3.10) I simply love this camera!

The problem illustrated in Example 3.9 is even more challenging. In the reviews, people tend to describe the setting in which they have a good or bad experience with a product. It happens quite often that in this setting description, the product name or other attributes are mentioned, which are in turn also extracted as opinion targets by the algorithm. Neither of the algorithms have a limit on how many opinion targets to extract for a given sentence. While this makes sense for enumerations or sentences which contain several opinions, it also leads to the mentioned problem and subsequently a low precision during the opinion extraction.

**Errors in Opinion Expression Identification**

As shown in Table 3.9, the decrease of both precision and recall regarding the opinion targets is quite substantial, when the MPQA lexicon is employed for the identification of opinion expressions / sentences. In the following, we will evaluate the precision and recall of the opinion expression identification using the MPQA lexicon. We will evaluate the opinion expression identification in two settings: In the first setting, we will require that the opinion expressions identified by the MPQA lexicon exactly match the gold standard annotations. This is equivalent to the evaluation strategy of the opinion targets and the results of this setting are shown in the three leftmost result columns of Table 3.13. In the second setting, we employ a lenient matching using the following boundary criteria: If there is a word overlap of at least one between the opinion expression as identified with the MPQA lexicon and the gold standard, we count it as a match. Otherwise, it is a non-match. The results of this evaluation strategy are shown in the three rightmost result columns of Table 3.13.

As shown in Table 3.13, the precision of the opinion identification using the MPQA lexicon is very low. We have performed a quantitative error analysis and have listed the most frequent false positives regarding opinion expressions as identified with the MPQA lexicon in Table 3.14.

As shown in Table 3.14, there are some words which frequently occur as false positives across all domains: The size and quantity describing adjectives "little" and "long" are very prominent in this category. The MPQA lexicon lists them both as negative opinion expressions, but in our datasets these words frequently occurred while not expressing any opinion. This might either be in phrases as "a long time

Table 3.13: Results Of Opinion Expression Extraction with MPQA

| | Exact Match | | | Partial Overlap | | |
|---|---|---|---|---|---|---|
| **Dataset** | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| movies | 0.116 | 0.524 | 0.190 | 0.129 | 0.582 | 0.211 |
| web-services | 0.144 | 0.361 | 0.206 | 0.154 | 0.385 | 0.220 |
| cars | 0.318 | 0.209 | 0.252 | 0.354 | 0.232 | 0.281 |
| cameras | 0.322 | 0.242 | 0.276 | 0.355 | 0.266 | 0.304 |

Table 3.14: Most Frequent False Positives in Opinion Expression Identification with the MPQA Lexicon

| **Dataset** | **Top Ten False Positives** |
|---|---|
| movies | plot, star, long, little, understand, truly, especially, fantasy, evil, honest |
| web-services | support, long, content, especially, extemely, serious, clearly, opportunity, truly, forget |
| cars | little, want, long, especially, plenty, extremely, throttle, patriot, truly, support |
| cameras | want, little, long, especially, sensitivity, serious, truly, creative, support, forget |

ago", factual descriptions e.g. "16 feet long" or generic characterizations which are not expressions of opinions as in:

(3.11) After our email ring was formed was when I delved a *little* deeper into e-groups and took a look around.

An approach for the identification of the polarity of such target-specific opinion expressions has been presented by Fahrni and Klenner (2008). It is to investigate whether their algorithm can be employed to also identify expressions as shown in the examples above, in which no opinions are expressed.

Furthermore, the inclusion of adverbs e.g. "especially", "plenty", "extremely", "clearly" leads to many false positives during the opinion expression identification on the datasets which we employ. Opinion annotation studies often differentiate between *modifiers* and opinion expressions. Terms as the adverbs mentioned above are typical examples of such modifiers. They do not express an opinion themselves, but are used in order to strengthen or weaken the opinion transported by the opinion expression they refer to.

We also observe some words which have a domain-specific meaning, which do not express an opinion in their respective domain. These are words as "fantasy", which is used to describe a genre as in "fantasy movie" in the movies domain, "sensitivity"

which occurs in "light sensitivity" - an attribute of a camera, analogous to "throttle" which is an attribute of a car. For such cases a relatively simple lexicon-based approach for the identification of opinion expressions is not sufficient, however more sophisticated approaches, e.g. based on machine learning, exist and represent the state-of-the-art for this task (Li and Zong, 2008; Choi and Cardie, 2009; Jijkoun et al., 2010).

The recurrence of the words "support" and "plot" as false positives can also be considered as a problem of domain adaptation: The reviews from the "web-services", "cars" and "cameras" datasets all deal with entities for which a "customer support" exist. This attribute is frequently discussed in the reviews, but "support" obviously does not express an opinion in this case. However, during the creation of the MPQA lexicon, also newswire discussing politics was employed. In the domains surrounding politics, the words "support" and "plot" primarily have a different meaning than in the datasets on which we evaluate.

The recall is also very low except on the "movies" dataset. We also observe that the lenient evaluation strategy which allows a partial match hardly increases precision and recall on all datasets. Table 3.15 gives an overview of the top 10 most frequent false negatives for each dataset.

Table 3.15: Most Frequent False Negatives in Opinion Expression Identification with the MPQA Lexicon

| **Dataset** | **Top Ten False Negatives** |
|---|---|
| movies | classic, overrated, powerful, original, strong, poor, fine, unique, top, predictable |
| web-services | easy, helpful, slow, simple, quick, easy to use, useful, fast, problem, easy to navigate |
| cars | new, comfortable, problem, unique, difficult, smooth, powerful, easy, expensive, quiet |
| cameras | new, fast, large, small, compact, high, easy, problem, more, advanced |

It is notable that apart from two words ("overrated", "more") and the two multiword expressions ("easy to use", "easy to navigate") all of the most frequent false negatives can be found in the MPQA lexicon, however they are labeled as weak subjectivity clues. A solution would be to also consider the weak subjectivity clues during the opinion expression identification, but given that the precision of the opinion expression identification is already very low when only the strong subjectivity clues are employed, the effects on the overall performance are likely to be negative. Especially in the "web-services", "cars" and "cameras" datasets we observe that there are many opinion expressions which are very context dependent. Words such as "new", "high", "quiet", "small" , "compact" and "large" are also frequently used in a non-opinionated context.

It is surprising that the quite sophisticated MPQA lexicon, which has been successfully employed in previous research many times, yields such a low precision and recall in the opinion expression identification on the datasets we employ in our

experiments. These results explain the poor performance of the opinion target extraction we observed in Setting IV (Table 3.9). The datasets we employ are also quite challenging regarding the opinion expression identification. The prior work in the field of opinion target identification or sentiment analysis indicates that supervised approaches often outperform lexicon-based approaches (Breck et al., 2007; Johansson and Moschitti, 2010). As mentioned above, such supervised approaches can address the challenge of domain-specific opinion expressions. Since the datasets we employ provide the training data required for supervised approaches, it would be interesting to investigate such an approach and subsequent effects on the opinion target extraction in future work.

**Errors in Extraction with Nearest Noun Phrase Heuristic**

In the following, we will show examples of typical errors which we observed for the *nearest noun phrase* heuristic. For the sake of readability and clarity, we selected relatively short example sentences. However, given that the average sentence length in each dataset is 17 tokens or more, the problems with the *nearest noun phrase* heuristic shown below tend to become even worse if the sentences get longer and therefore more complex. Another challenge which we face with this heuristic is, that the algorithm only selects one target per opinion expression. If there is only one target in a sentence, the selection of a false target will result in both a false positive and a false negative (unless there are no noun phrase candidates available in a sentence e.g. due to an error in the preprocessing).

We clustered the sentences containing errors around a set of recurring problems which we observed in our error analysis. Correctly extracted opinion targets (true positives) are underlined, false positives are highlighted by a wavy underline, false negatives are highlighted by a dotted underline and the opinion expressions are shown in boldface.

**Intermediate Phrases / Clauses:**

It is quite common that a prepositional phrase or a relative clause is placed between the opinion expression and the opinion target as shown in Examples 3.12 to 3.14:

(3.12) The final scene in the Roman Coliseum is **ridiculous** and descends to the level of a Cecil B. De Mille circus.

(3.13) And James Whitmore's portrail [sic] of an elderly inmate Brooks is **moving**.

(3.14) Another **big feature** at Ecircles is the game area.

The algorithm will then always falsely select the closer noun phrase in the prepositional phrase or relative clause. This is a clear limitation of a word distance based heuristic. A more sophisticated grammatical analysis e.g. with dependency parsing is required here. By employing a dependency parser, we can identify that the adjective "ridiculous" in Example 3.12 modifies the noun "scene", or that the predicate adjective "moving" in Example 3.13 refers to the nominal "portrayal". Note that although "portrayal" is misspelled in Example 3.13, the complete phrase could be extracted since the TreeTagger assigns a noun tag to unknown words.

**Conjunctions:**

In the following, we will show some example sentences in which a correct target is found, but some targets are also missing:

(3.15) Most notably, the Hyundai engineers have done an **excellent job** with the chassis and suspension.

(3.16) Tom Hanks is **superb**, as are Gary Sinise, Mykelti Williamson and Sally Field.

As shown in Examples 3.15 and 3.16, the heuristic typically fails to extract some targets if one opinion expression refers to several targets, e.g. in an enumeration. This could be solved by extending the heuristic to extract adjacent noun phrases to the selected candidate, if they are connected with a coordinating conjunction. However, as shown in Example 3.16, there are sentences in which this extension is not sufficient to capture all targets.

## 3.3 Enriching the LRT with Encyclopedic Information

The Likelihood Ratio Test (LRT) presented in Section 3.1.2 has been successfully employed for two tasks which are closely related to the extraction of opinion targets:

1. The extraction of product features in an opinion mining task (Yi et al., 2003; Hu and Liu, 2004a), which has gained importance due to the popularity of Web 2.0 and the massive amounts of customer reviews e.g. written on popular e-commerce platforms. Here, identified product features are used e.g. to create feature-oriented summaries of customer review collections, or as the basis for extracting opinions about features.

2. Keyphrase extraction (Turney, 2000; Tomokiyo and Hurst, 2003) aims at identifying the most relevant words and phrases in a document collection, and is one of the pervasive tasks in natural language processing. It is very closely related to Information Extraction tasks such as Terminology extraction as discussed in Chapter 1.

Keyphrase extraction and product feature extraction mainly differ in their definition of "relevance". In keyphrase extraction, the goal is to identify those words[9] in a given document which best describe its topic by distinguishing it from documents with different topics. Individual mentions of the same word are not considered. In product feature extraction, on the other hand, the goal is to extract all mentions of features for a given product. At the same time, it is important to only extract features of the product under review, and not of any other products mentioned e.g. in comparisons.

In (Ferreira et al., 2008), we present a comparative evaluation of the approaches by Hu and Liu (2004a) and Yi et al. (2003), while evaluating the product feature extraction independently of opinion detection. We identified two limitations of the LRT, which have also emerged in the error analysis of our opinion target extraction in 3.2.3:

[9]We use *words* here to cover both single words as well as multiword expressions.

1. It often fails to identify **rare** opinion targets.

2. It also often fails to identify opinion targets which are frequent in the general vocabulary (e.g. "weight", "speed" and "option").

In the following, we will introduce LRT$_{wiki}$, our extension of the Likelihood Ratio Test algorithm, which addresses the above limitations by enriching the LRT with encyclopedic information drawn from Wikipedia. In LRT$_{wiki}$, Wikipedia is employed as a general-purpose source of domain knowledge. We analyze the performance of LRT$_{wiki}$ in three different scenarios: In the first scenario, we employ the algorithm for product feature extraction as in (Yi et al., 2003) and (Ferreira et al., 2008), in the second scenario we employ it for keyphrase extraction as in (Tomokiyo and Hurst, 2003), and in the third scenario we will employ it for the extraction of opinion targets as in Section 3.2.

### 3.3.1 LRT and LRT$_{wiki}$

In their application of the LRT to product feature extraction, Yi et al. (2003) and Ferreira et al. (2008) report high precision but low recall. In (Ferreira et al., 2008) we observed that the LRT typically misses product features that have a low frequency in the on-topic document collection $D_+$, e.g. because only very few customers comment on them - even if they do not occur in the off-topic document collection $D_-$ at all. This problem also manifested itself in the task of opinion target extraction, as shown in the target distribution histograms in Section 3.2.3. In addition, the LRT also misses terms which are both product features and general vocabulary items, such as "speed", "option" and "flexibility".

LRT$_{wiki}$ aims at improving the ranking of candidate terms of two problematic candidate classes:

1. Candidate terms which occur in the on-topic document collection $D_+$ with low frequency and not at all or with a low frequency in the off-topic document collection $D_-$

2. Candidate terms which occur both in the on- and off-topic document collections with medium or high frequency

The central idea of LRT$_{wiki}$ is to employ a comprehensive domain-specific corpus containing the terminology typically used in the current domain as the source of additional on-topic datasets, and to modify the calculation of the LRT algorithm to take advantage of this new domain-specific corpus. This corpus is created on the basis of Wikipedia. We chose the free online encyclopedia Wikipedia for two reasons:

1. Due to its broad coverage, we can expect it to contain articles about many topics. Thus the method will be easily scalable to new domains.

2. Due to the encyclopedic style of Wikipedia articles, they tend to focus on a single topic, and normally do not contain irrelevant information. In our setting, irrelevant information would e.g. be if in a product review the customer drifts off-topic and discusses something unrelated to the product under review.

Since our goal is to extract an additional corpus about the pre-defined topic(s) dealing with the document collection to be analyzed, we assume the topic to be known in advance. We then query Wikipedia in order to retrieve one article for each topic as the seed for retrieving the new "on-topic" datasets. The Wikipedia-based document collection for each topic is built by extracting the categories to which the seed article belongs and then extracting all articles found in these categories. We performed a simple ad-hoc filtering only: We ignored all subcategories of "Wikipedia administration", since they do not contain any information relevant for us, and all categories with more than 200 articles, since we regard them as too broad. Algorithm 2 formally describes the crawling and corpus creation.

**Algorithm 2** Wikipedia Corpus Creation

```
Set seedCategories = A_Seed.getCategories()
Set relatedArticles = new Set
for all Category C_i in seedCategories do
  if C_i.getNumChildren() < 200 then
    relatedArticles.add(C_i.getChildren())
  end if
end for
return relatedArticles
```

### Enhancing the LRT Algorithm

The new Wikipedia content provides an additional document collection $D_W$ on the basis of which we can calculate $C_{13}$ and $C_{23}$ for a given term $T$, which are defined in Table 3.16.

Table 3.16: Extended Contingency Table for $\text{LRT}_{wiki}$

| | $\mathbf{D}_+$ | $\mathbf{D}_-$ | $\mathbf{D}_W$ |
|---|---|---|---|
| $\mathbf{T}$ | $C_{11}$ | $C_{12}$ | $C_{13}$ |
| $\overline{\mathbf{T}}$ | $C_{21}$ | $C_{22}$ | $C_{23}$ |

With these new values we modify the calculation of the original Likelihood Ratio Score $lr$ as shown in Equation 3.17:

$$
\begin{aligned}
lr_{mod} =& (C_{11mod} + C_{21}) \log(r) + (C_{12mod} + C_{22}) \log(1-r) - C_{11mod} \log(r_1) \\
& - C_{12mod} \log(1-r_1) - C_{21} \log(r_2) - C_{22} \log(1-r_2) \\
C_{11mod} =& \begin{cases} C_{11} + C_{13} & \text{if } C_{11} < t_1 \text{ and } C_{12} < t_1 \\ C_{11} + C_{13} & \text{if } C_{11} > t_2 \text{ and } C_{12} > t_2 \end{cases} \\
C_{12mod} =& \begin{cases} 0 & \text{if } C_{11} < t_1 \text{ and } C_{12} < t_1 \\ \max(0, C_{12} - C_{13}) & \text{if } C_{11} > t_2 \text{ and } C_{12} > t_2 \end{cases}
\end{aligned}
\tag{3.17}
$$

The two thresholds $t_1$ and $t_2$ are used to set the boundaries for terms with low frequency ($t_1$) and terms with medium or high frequency ($t_2$).

### 3.3.2 Datasets

We will now describe the two additional datasets which we employ to evaluate our $\text{LRT}_{wiki}$ algorithm in the tasks of product feature extraction and keyphrase extraction next to the datasets we introduced in Chapter 2.

#### Data for Opinion Target Extraction

For our evaluation of the $\text{LRT}_{wiki}$ algorithm in the task of opinion target extraction, we employed the same datasets as presented in Chapter 2.

#### Data for Product Feature Extraction

We employ datasets of customer reviews for five products, collected from Amazon.com and C|net.com as described by Hu and Liu (2004a). These customer reviews focus on electronic products: two digital cameras, a DVD player, an MP3 player and a cell phone. Table 3.17 presents descriptive statistics about each dataset.

Table 3.17: Product Review Datasets

| Dataset | Documents | Sentences |
|---|---|---|
| Digital camera 1 (DC1) | 45 | 597 |
| Digital camera 2 (DC2) | 34 | 346 |
| Cell phone (CP) | 41 | 546 |
| MP3 player (MP3) | 95 | 1716 |
| DVD player (DVD) | 99 | 739 |

#### Annotation Scheme by Hu and Liu (2004a) and its Revision

Hu and Liu (2004a) define a product feature as a characteristic of the product which customers have expressed an opinion about, where an opinion is a statement which explicitly characterizes a feature in a positive or negative manner. Their annotation consists of the product feature(s) mentioned in the current sentence, where a feature is only annotated as such if an opinion is stated about it. For instance in the sentence:

(3.18) at the same time, i wanted my wife to not be intimidated by knobs and buttons.

no features are annotated, although the product features “knobs” and “buttons” are mentioned. Since we also focus on the product feature extraction step, we additionally annotated features in neutral sentences which contain product features, such as sentence 3.18. At the end of our annotation phase, all product features in the reviews will be annotated, both in opinion as well as non-opinion bearing sentences.

In our revised annotation scheme, each entity to be annotated as a feature must satisfy one of the following criteria:

- *Part-of* relationship with the product the document is about; for example in the domain of digital cameras "battery" would be annotated as a feature of a camera.
- *Attribute-of* relationship with the product; for example "weight" and "design" would be considered as attributes of a camera.
- *Attribute-of* relationship with a known feature of the product of the document; for example "battery life" would be considered an *attribute of a feature* of the camera, specifically an attribute of the "battery".

For example, in the sentence:

(3.19) the lens is visible in the viewfinder when the lens is set to the wide angle , but since i use the lcd most of the time , this is not really much of a bother to me.

the features "lens", "viewfinder" and "lcd" are annotated in our annotation scheme, but not by Hu and Liu (2004a).

Table 3.18 presents comparative statistics based on the data annotated according to the original and revised annotation schemes. The second column gives the total number of distinct product features annotated in each set of documents of the review data. Column 4 shows the number of distinct features found in the revised annotation. Columns 3 and 5 contain the number of annotated features where every instance of a product feature is counted.

Table 3.18: Number of Features in Original and Revised Annotation

| | **Original Annotation** | | **Revised Annotation** | |
|---|---|---|---|---|
| **Dataset** | Distinct | Total | Distinct | Total |
| DC1 | 99 | 257 | 161 | 594 |
| DC2 | 74 | 185 | 120 | 340 |
| CP | 109 | 310 | 140 | 471 |
| MP3 | 180 | 736 | 231 | 1031 |
| DVD | 110 | 347 | 166 | 519 |

We observe that the revised annotation contains far more features than the original annotation. This was to be expected since we annotated features irrespectively of an opinion being expressed about them or not.

The revised annotation was originally performed by just one annotator. Since a verification of the reliability of the annotation is important (e.g. as an upper bound for the evaluation of the product feature extraction task), we re-annotated a subset of the corpus in a controlled manner. First, we randomly selected 60 sentences from each of the five product review sets. Then, we had two human subjects annotate

them following the guidelines presented at the beginning of the current Section. Due to the skewed class distribution (the vast majority of terms in product reviews are *not* product features), we calculated precision, recall, and F-Measure (instead of e.g. $\kappa$ (Cohen, 1960)) on the overlap between the two annotators. The overlap was calculated rather strictly by considering only exact matches in the product feature annotation. The results of the annotation overlap measurements are shown in Table 3.19. We measured the overlap in precision and recall, however in a two annotator setting one annotator's precision is the other annotator's recall value, hence we only report F-Measure in the table.

Table 3.19: Annotation Overlap for Product Feature Mentions

| **Dataset** | **Sentences** | **Words** | **Features** | **F-Measure** |
|---|---|---|---|---|
| DC1 | 60 | 980 | 67 | 0.736 |
| DC2 | 60 | 1029 | 69 | 0.747 |
| CP | 60 | 1001 | 63 | 0.825 |
| MP3 | 60 | 830 | 46 | 0.745 |
| DVD | 60 | 883 | 52 | 0.477 |

An analysis of the annotation overlap shows that product feature extraction is not a trivial task. Although the inter-annotator agreement is decent on the cellphone dataset with an F-Measure of 0.825, the agreement on both digital camera datasets and the mp3-player dataset is just between 0.736 and 0.747. The F-Measure on the DVD player dataset is particularly low. An analysis of the cases of disagreement has revealed, that this is due to excessive usage of abbreviations regarding the product in this document collection (e.g. referring to the product with just its model number) and some disagreement regarding their annotation.

### Data for Keyphrase Extraction

The data we employ in the keyphrase extraction experiments is originally from the DUC2001 dataset (Over, 2001). The corpus consists of 309 news articles with keyphrases annotated by Wan and Xiao (2008). The articles cover 30 different news topics and have an average length of 740 words. The annotation involved two annotators, who were allowed to select a maximum of 10 distinct keyphrases per document. Wan and Xiao (2008) report an inter-annotator agreement of 0.70 $\kappa$[10]. After the annotation, the annotators created the final gold standard by resolving conflicting annotations in a discussion. The average number of keyphrases per document is 8.08, and the average number of words per keyphrase is 2.09 (Wan and Xiao, 2008).

From the entire DUC2001 dataset, which can be considered as a collection of 30 subcorpora regarding different topics, we selected the two largest subcorpora for our evaluation. Each of these two subcorpora (DUC IDs: d06a & d34f) contains 16

[10]Cohen (1960) defines $\kappa = \frac{Pr(a)-Pr(e)}{1-Pr(e)}$, where $Pr(a)$ is the observed agreement between the two annotators, and $Pr(e)$ is the probability of chance agreement.

documents, which are each employed in an experiment as the "on-topic" document collections $D_+$. The tasks of our evaluation are hence two keyphrase extraction runs, once on the d06a corpus and once on the d34f corpus.

### 3.3.3 Experiments and Results

As outlined in Sections 3.1.2 and 3.3, the LRT has been successfully applied for several term extraction tasks as in (Yi et al., 2003; Ferreira et al., 2008; Tomokiyo and Hurst, 2003). In our evaluation, we aim at analyzing our modifications to the algorithm in three tasks: Opinion target extraction analogous to our experiments in Section 3.2, product feature extraction as in (Ferreira et al., 2008) and keyphrase extraction as in (Tomokiyo and Hurst, 2003). The opinion target extraction will be evaluated on the datasets presented in Chapter 2 and for the other two tasks, we will present the evaluation dataset in Section 3.3.2.

In Figure 3.6, we present a conceptual overview of the different datasets employed and the different ranking approaches of the LRT and LRT$_{wiki}$ which we will evaluate in our experiments.

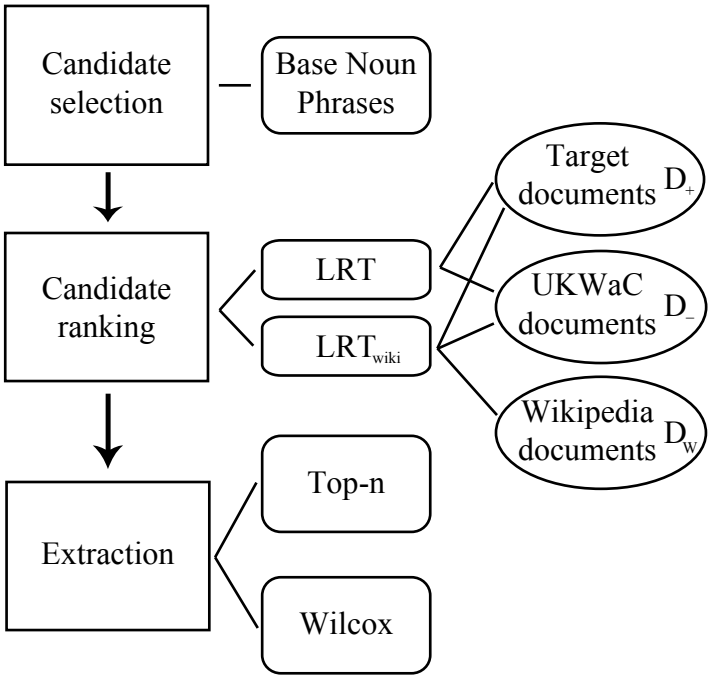



Figure 3.6: Term Extraction Architecture

#### Wikipedia Corpus $D_W$ Creation

As defined in Algorithm 2, the Wikipedia crawling requires a seed article $A_{Seed}$ to start from. For the datasets presented in Chapter 2, we used the domain descriptions as provided in the respective papers (Zhuang et al., 2006; Kessler and Nicolov, 2009; Toprak et al., 2010) as seed articles for our Wikipedia crawling ("movies", "cars", "cameras", "web-services")[11]. For each of these topics there is either a Wikipedia

[11]Naturally each in the singular form, as this is how encyclopedia articles are named.

article with the same title or an automatic redirect page. For the product classes from the Hu and Liu (2004a) dataset, we used the names provided in their paper (*digital camera*, *mp3 player* → *Digital audio player*, *cell phone*, *dvd player*, *mp3 player*). For the DUC data, we read the documents and inferred the topics "police brutality" (d06a), for which Wikipedia contains an article, and "atlantic hurricanes" (d34f), which is redirected to "North Atlantic tropical cyclone". For the datasets employed in the evaluation of the opinion target extraction we used the domain descriptions of the datasets (*movie* → *film*, *web service*, *car* → *automobile*, *digital camera*). Table 3.20 provides an overview of the tasks we evaluate the algorithms on, the thereby employed datasets and the corresponding seed article which we start from during the Wikipedia corpus creation.

Table 3.20: Overview of Evaluation Task, Employed Datasets and Seed Articles for $\text{LRT}_{wiki}$

| **Evaluation Task** | **Dataset** | **Wikipedia Seed Article** $A_{Seed}$ |
|---|---|---|
| Opinion Target Extraction | movies | film |
| | web-services | web-service |
| | cars | automobile |
| | cameras | digital camera |
| Product Feature Extraction | DC1, DC2 | digital camera |
| | CP | cell phone |
| | MP3 | digital audio player |
| | DVD | dvd player |
| Keyphrase Extraction | d34f | north atlantic tropical cyclone |
| | d06a | police brutality |

The article pages retrieved in this manner were then automatically cleaned of all Wikipedia markup, metadata, references, and hyperlinks by manually defined regular expressions. Details on the retrieval and cleaning are provided in Appendix B. The data we retrieved is extracted from a Wikipedia dump from February 2007. Some statistics about the resulting data are given in Table 3.21.

Analogous to our experimental setup in Section 3.2.1, we employ the same 600 randomly selected documents from the UKWaC British English web corpus (Ferraresi et al., 2008) as an off-topic corpus ($D_-$).

For $\text{LRT}_{wiki}$, we set $t_1$ to 5, which was empirically defined in order to reflect a threshold under which we consider a term to be rare. Likewise, the threshold $t_2$ was set to 10, meaning we consider terms which are found more than 10 times to be frequently occurring. These thresholds are optimal to the corpora we experimented with, while smaller or larger values might make sense for different input corpora $D_+$.

Table 3.21: Content Retrieved from Wikipedia for $D_W$ Datasets

| **Wikipedia Seed Article** | **Retrieved Articles** | **Tokens** |
|---|---|---|
| film | 204 | 297744 |
| web service | 85 | 80881 |
| automobile | 54 | 76522 |
| digital camera | 263 | 161459 |
| cell phone | 250 | 204410 |
| digital audio player | 64 | 79099 |
| dvd player | 100 | 99898 |
| north atlantic tropical cyclone | 403 | 459046 |
| police brutality | 166 | 127216 |

### Opinion Target Extraction

Since the LRT$_{wiki}$ algorithm can be provided with the same input as the original LRT and produces the same output, we employ the identical experimental setup as presented in 3.2.1. Since our error analysis in Section 3.2.3 has shown that the MPQA lexicon yields inadequate results in the opinion expression identification, we will only evaluate LRT$_{wiki}$ in Settings I and II. Table 3.22 shows the results of the LRT$_{wiki}$ algorithm in the task of opinion target extraction.

Table 3.22: Results LRT$_{wiki}$ for Opinion Target Extraction

| | LRT$_{wiki}$ | | | **LRT**$_{wiki}$ + *nearest noun phrase* | | |
|---|---|---|---|---|---|---|
| **Dataset** | Precision | Recall | F-Measure | Precision | Recall | F-Measure |
| movies | 0.307 | 0.728 | 0.432 | 0.306 | 0.732 | 0.432 |
| web-services | 0.292 | 0.485 | 0.364 | 0.291 | 0.505 | 0.369 |
| cars | 0.203 | 0.363 | 0.260 | 0.205 | 0.381 | 0.266 |
| cameras | 0.232 | 0.399 | 0.293 | 0.235 | 0.421 | 0.302 |

If we compare the three leftmost result columns to the results of the original LRT from Table 3.6, we observe that LRT$_{wiki}$ yields a slight increase of recall on all datasets. However, the decrease of precision in the target extraction outweighs the increase of recall, leading to a slight decrease regarding F-Measure.

We can compare the results in the three rightmost results columns with the results of the "LRT + nearest noun phrase" configuration of Table 3.7. In this configuration, the trend of an increase in recall at the expense of precision is also observable. However, here the gains of recall outweigh the loss in precision, leading to a slightly higher (or unchanged) F-Measure on the four datasets. The improvements

are however not statistically significant.[12]

### Product Feature Extraction

Table 3.23 shows the results obtained in (Ferreira et al., 2008) and the results of our experiments. In the table section "LRT in (Ferreira et al., 2008)" we present the results of the original LRT as presented in (Yi et al., 2003). The table section "LRT Wilcox Threshold" shows the effects of the threshold calculation based on Wilcox (2001), as introduced in 3.2.1. Finally, the section "$\text{LRT}_{wiki}$ Wilcox Threshold" shows the results obtained by employing the threshold calculation based on Wilcox (2001) and $\text{LRT}_{wiki}$. Following our evaluation strategy in (Ferreira et al., 2008), we perform an evaluation on each mention of a product feature, comparing the lowercased and lemmatized forms of the automatically extracted features with those in the gold standard. Again, an exact match regarding the boundaries of a product feature is required for a true positive, partial overlap will be considered a false extraction. When comparing the results of "LRT Wilcox Threshold" with "LRT in (Ferreira et al., 2008)", we observe a constant improvement in precision and recall. This shows that the threshold calculation strategy for the extraction, as introduced in Section 3.2.1, is also an important aspect of the LRT which might deserve further research. The recall slightly decreases when comparing the LRT and $\text{LRT}_{wiki}$ on two of the datasets (DC2, MP3). However, when comparing the original LRT with $\text{LRT}_{wiki}$, we observe that the concurrent gains in precision outweigh the losses in recall, leading to an overall significantly higher F-Measure.[13]

### Keyphrase Extraction

As we are interested in a state-of-the-art approach for unsupervised keyphrase extraction, we employ the TextRank system (Mihalcea and Tarau, 2004). We follow Mihalcea & Tarau by selecting only adjectives and nouns as candidate terms. The matching is done in a greedy fashion on the terms' POS tags with the following regular expression: $(JJ|JJR|JJS)^*(NN|NNS|NP|NPS)^+$. Greedy matching makes sure that only the longest matching phrases in a sentence are selected as candidates. This matching strategy is based on the observation that *complete* noun phrases are typically annotated as keyphrases in the DUC dataset. For example "accidental shooting death" is annotated as a keyphrase and not just "shooting death" or "death". Contrary to product features, there is no clear-cut definition of what is and what is not regarded as a keyphrase for a document. Therefore, during our evaluation, we did not employ a threshold like in 3.3.3. Alternatively, we evaluate precision, recall and F-Measure of the *top-n* extracted keyphrases. As a baseline system, we employ TextRank in its default configuration. All keyphrases are lemmatized and lowercased before comparison and again we require an exact boundary match for a true positive during the extraction. A partially extracted keyphrase will be counted as a false positive / false negative respectively. When employing $\text{LRT}_{wiki}$, we use the same thresholds $t_1$ and $t_2$ as described in Section 3.3.3. We

[12] Significance of improvements is tested using a paired two-tailed t-test with $p \leq 0.05$.

[13] Significance of improvement in F-Measure is tested using a paired one-tailed t-test and $p \leq 0.05$ (*), $p \leq 0.01$ (**), and $p \leq 0.005$ (***).

Table 3.23: Results $LRT_{wiki}$ for Product Feature Extraction

| | **LRT in (Ferreira et al., 2008)** | | |
|---|---|---|---|
| **Dataset** | Precision | Recall | F-Measure |
| DC1 | 0.671 | 0.495 | 0.570 |
| DC2 | 0.634 | 0.347 | 0.449 |
| CP | 0.659 | 0.459 | 0.541 |
| MP3 | 0.339 | 0.408 | 0.370 |
| DVD | 0.506 | 0.243 | 0.328 |
| | **LRT Wilcox Threshold** | | |
| DC1 | 0.750 | 0.513 | 0.609 |
| DC2 | 0.800 | 0.485 | 0.604 |
| CP | 0.579 | 0.535 | 0.556 |
| MP3 | 0.513 | 0.665 | 0.579 |
| DVD | 0.633 | 0.416 | 0.502 |
| | $LRT_{wiki}$ Wilcox Threshold | | |
| DC1 | 0.760 | 0.574 | 0.654* |
| DC2 | 0.875 | 0.474 | 0.615* |
| CP | 0.813 | 0.544 | 0.651* |
| MP3 | 0.560 | 0.661 | 0.606* |
| DVD | 0.667 | 0.458 | 0.543* |

evaluate the top-n keyphrases ($1 \leq n \leq 10$) on the two datasets each containing 16 documents as described in Section 3.3.2. The results of the keyphrase extraction evaluation are shown in Table 3.24.

When comparing the results of the keyphrase extraction based on "LRT Wilcox Threshold" and "$LRT_{wiki}$" with TextRank as a baseline, we observe that on the d06a dataset both LRT versions perform considerably better and on the d34f dataset considerably worse than the TextRank system. However, the performance of TextRank on the d34f dataset is much better than its average on the entire DUC2001 dataset: TextRank yields an overall F-Measure of 0.132 at 10 extracted keyphrases on the entire DUC2001 dataset,[14] while on the d34f dataset it reaches an F-Measure of 0.319 at 10 extracted keyphrases. Apart from the configurations when only the top 1 and 2 keyphrases are extracted on the d34f dataset, $LRT_{wiki}$ significantly increases the F-Measure[15] in the keyphrase extraction task.

[14]We obtained this result in a separate experiment.

[15]Significance of improvement in F-Measure is tested using a paired one-tailed t-test and $p \leq 0.05$ (*), $p \leq 0.01$ (**), and $p \leq 0.005$ (***). In this experiment, we can indicate the significance directly via the asterisks, as we do not compare any results across different Tables with different configurations.

Table 3.24: Results Keyphrase Extraction

| | | **TextRank** | | | **LRT** | | | **LRT$_{wiki}$** | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Dataset** | **n** | Prec. | Rec. | F-M. | Prec. | Rec. | F-M. | Prec. | Rec. | F-M. |
| **d06a** | 1 | 0.188 | 0.029 | 0.051 | 0.000 | 0.000 | 0.000 | 0.062 | 0.010 | 0.017*** |
| (16 docs) | 2 | 0.125 | 0.039 | 0.060 | 0.094 | 0.030 | 0.045 | 0.375 | 0.119 | 0.180*** |
| | 3 | 0.083 | 0.039 | 0.053 | 0.250 | 0.119 | 0.161 | 0.333 | 0.158 | 0.215*** |
| | 4 | 0.094 | 0.059 | 0.072 | 0.281 | 0.178 | 0.218 | 0.328 | 0.208 | 0.255*** |
| | 5 | 0.088 | 0.069 | 0.077 | 0.250 | 0.198 | 0.221 | 0.325 | 0.257 | 0.287*** |
| | 6 | 0.073 | 0.069 | 0.071 | 0.250 | 0.238 | 0.244 | 0.312 | 0.297 | 0.305*** |
| | 7 | 0.071 | 0.078 | 0.075 | 0.232 | 0.257 | 0.244 | 0.295 | 0.327 | 0.310*** |
| | 8 | 0.070 | 0.088 | 0.078 | 0.234 | 0.297 | 0.262 | 0.273 | 0.347 | 0.306*** |
| | 9 | 0.069 | 0.098 | 0.081 | 0.222 | 0.317 | 0.261 | 0.250 | 0.356 | 0.294*** |
| | 10 | 0.075 | 0.118 | 0.092 | 0.219 | 0.347 | 0.268 | 0.238 | 0.376 | 0.291*** |
| **d34f** | 1 | 0.467 | 0.053 | 0.096 | 0.062 | 0.008 | 0.014 | 0.062 | 0.008 | 0.014 |
| (16 docs) | 2 | 0.500 | 0.115 | 0.186 | 0.062 | 0.015 | 0.024 | 0.062 | 0.015 | 0.024 |
| | 3 | 0.500 | 0.168 | 0.251 | 0.042 | 0.015 | 0.022 | 0.062 | 0.023 | 0.033*** |
| | 4 | 0.448 | 0.198 | 0.275 | 0.062 | 0.030 | 0.041 | 0.078 | 0.038 | 0.051*** |
| | 5 | 0.431 | 0.237 | 0.305 | 0.100 | 0.061 | 0.075 | 0.150 | 0.091 | 0.113*** |
| | 6 | 0.393 | 0.252 | 0.307 | 0.115 | 0.083 | 0.096 | 0.167 | 0.121 | 0.140*** |
| | 7 | 0.396 | 0.290 | 0.335 | 0.125 | 0.106 | 0.115 | 0.170 | 0.144 | 0.156*** |
| | 8 | 0.374 | 0.305 | 0.336 | 0.148 | 0.144 | 0.146 | 0.172 | 0.167 | 0.169*** |
| | 9 | 0.350 | 0.313 | 0.331 | 0.139 | 0.152 | 0.145 | 0.167 | 0.182 | 0.174*** |
| | 10 | 0.325 | 0.313 | 0.319 | 0.138 | 0.167 | 0.151 | 0.169 | 0.205 | 0.185*** |

### Error Analysis

As evident from Tables 3.23 and 3.24, LRT$_{wiki}$ consistently and significantly[16] improves F-Measure in the tasks of product feature extraction and keyphrase extraction. In the task of opinion target extraction, the improvements are not significant. In the following, we perform an error analysis in the three applications separately.

**Opinion Target Extraction Error Analysis:**

A manual inspection of the target candidate lists of the LRT$_{wiki}$ approach has shown that they contain slightly more candidates than the LRT in the original configuration. The newly added candidates seem to be of good quality, which is also indicated by the gains in recall on all datasets. However, the slightly bigger candidate lists also introduce more false positives during the target extraction. We attribute this to the challenge of candidate terms frequently occurring in opinion

[16]Significance of improvement in F-Measure is tested using a paired one-tailed t-test and $p \leq 0.05$ (*), $p \leq 0.01$ (**), and $p \leq 0.005$ (***). In this experiment, we can indicate the significance directly via the asterisks, as we do not compare any results across different Tables with different configurations.

sentences while not being the actual opinion targets, as observed in our error analysis in Section 3.2.3.

**Product Feature Extraction Error Analysis:**

As shown in Table 3.23 the recall slightly decreases when comparing the LRT and LRT$_{wiki}$ on two of the datasets (DC2, MP3). The decrease of recall on some datasets can be explained as follows: A substantial amount of terms belonging to the specific vocabulary of the domain have $C_{11}$ and $C_{12}$ values smaller than $t_1$ and therefore receive a *boosting* from the new Wikipedia content. The boosting often pushes their $lr_{mod}$ values into the regions of other terms which have a $C_{11} > t_1$ and which would therefore typically be extracted as relevant. However, the boosting effect raises the overall average likelihood ratio, which we use to separate the relevant from the irrelevant terms. At the same time, there are typically quite a few terms which occur in almost every sentence (e.g. the product under review) and which therefore influence the standard deviation. In general, the boosting modification leads to a substantial increase in the average likelihood ratio, while hardly affecting the standard deviation. This leads to a slight increase in the threshold for the extraction of terms, with some relevant terms no longer reaching it. On the DC1 dataset, e.g. LRT$_{wiki}$ extracts the correct product features "control", "film", and "sensor", while the original LRT misses them. At the same time, using LRT$_{wiki}$, the correct features "external flash" and "lcd screen" do not reach the threshold any more while the original LRT extracts them.
This effect on the average likelihood ratio (which is even more pronounced for the standard deviation) is caused by the modification which aims to improve the extraction of relevant terms also occurring in the general language corpus e.g. "weight", "speed" or "option" for the camera domain. Such candidates typically have a rather high $C_{11}$ value, and due to the Wikipedia content their $C_{12}$ value is reduced, leading to a very high likelihood ratio, which in turn leads to a higher standard deviation.

The inclusion of the Wikipedia documents also exacerbates one of the issues mentioned in (Ferreira et al., 2008): If several reviews mention products of another manufacturer or a different model (e.g. in comparisons), the LRT (and also LRT$_{wiki}$) will extract them. Since the documents from Wikipedia are typically not limited to a single product, but rather to a product class, they tend to contain names of several different models and manufacturers. If such a name is mentioned in the "on-topic" documents, its likelihood ratio will be boosted due to our algorithm modification. This might lead to an extraction of such a model or manufacturer name which results in a false positive. Overall, however, LRT$_{wiki}$ still leads to a significant improvement over LRT regarding F-Measure.

**Keyphrase Extraction Error Analysis:**

As mentioned above, the TextRank algorithm consistently outperforms both LRT configurations on the d34f dataset. This is due to the fact that, with three to five words, the keyphrases on the d34f subset are rather long compared to those in the other document sets (overall average keyphrase length in words: 2.0). When employed in a keyphrase extraction task, both versions of the LRT have the limitation that the relevance of a term is calculated on the overall document collection. However, keyphrases are annotated with respect to their importance in individual

documents. Therefore, the LRT often fails to extract keyphrases which are only relevant for one document. This definition of relevance is different from the product feature extraction task, as terms are regarded as relevant over the entire document collection.

## 3.4 Chapter Summary

In this chapter we provided a comprehensive analysis of two state-of-the-art unsupervised algorithms for extracting opinion targets based on the Likelihood Ratio Test and on Association Mining. We have shown how the two algorithms perform in settings in which opinion expressions are identified at different levels of granularity and exactness. In our evaluation, the Likelihood Ratio Test based approach generally yielded better results. We have furthermore employed a non-statistical approach, which relies on identified opinion expressions for the extraction of opinion targets based on a nearest noun phrase heuristic. This heuristic outperformed both algorithms in a setting in which the opinion expressions were identified with perfect accuracy. However the automatic identification of individual opinion expressions is a challenging task.

We have also evaluated the performance of the three approaches mentioned above given that the opinion expressions are identified using a state-of-the-art lexicon of subjectivity clues. The performance of all approaches decreased considerably for which we have identified an inadequate accuracy regarding the opinion expression identification in our error analysis.

Our evaluation has shown that both the Association Mining-based approach as well as the Likelihood Ratio Test-based approach extract the opinion targets with a low precision. This is due to the fact that the highly ranked target candidates also occur in opinion sentences in which they are not the actual targets. The *nearest noun phrase* heuristic yields better results since it only extracts one target per opinion expression hence reducing the number of false positives.

The recall of the statistical approaches was relatively low on three out of four datasets. Our error analysis has revealed that especially in these three datasets a large share (up to 55%) of the opinion targets only occur up to five times. Such rare opinion targets are typically difficult to extract with statistical approaches, as they are ranked low regarding relevance in a corpus due to their low frequency. This phenomenon severely limits the recall of the target extraction. At the same time the *nearest noun phrase* heuristic did not yield considerably higher results regarding recall. This shows that even when the opinion expressions are provided, it is not trivial to select the corresponding target(s) in a given sentence. More sophisticated approaches which analyze the grammatical structure of a sentence, e.g. with dependency / constituent parsing, could be promising for future work.

Motivated by the insights gained in our error analysis, we presented $\text{LRT}_{wiki}$, an enhancement of the Likelihood Ratio Test, which makes use of encyclopedic documents retrieved from Wikipedia as an additional source of information for the extraction of e.g. rare, but at the same time relevant terms. We evaluated $\text{LRT}_{wiki}$ in the tasks of opinion target extraction, product feature extraction and keyphrase extraction. The enhanced algorithm leads to an improvement regarding the relevance

ranking of terms. Since Wikipedia is available in many languages and its content is very broad, it seems to be a well suited resource for extending a statistical method for information extraction tasks such as terminology or keyphrase extraction. In the opinion target extraction task, we could not reach any significant improvements with our algorithm $\text{LRT}_{wiki}$ over the original Likelihood Ratio Test. However, $\text{LRT}_{wiki}$ significantly improves over the original Likelihood Ratio Test in the tasks of product feature extraction and keyphrase extraction.

# Chapter 4

# Supervised Extraction of Opinion Targets

The extraction of opinion targets can be considered an instance of an information extraction (IE) task (Cowie and Lehnert, 1996). Conditional Random Fields (CRF) (Lafferty et al., 2001) have been successfully applied to several IE tasks in the past (Peng and McCallum, 2006). Compared to hidden Markov models, CRFs have the advantage of their conditional nature, which allows for a greater flexibility regarding the modeling of state transitions. The conditional design of CRFs enables a relaxation of the independence assumptions regarding the states required by hidden Markov models in order to guarantee for a controllable inference. A recurring problem, which arises when working with supervised approaches, concerns the domain portability. In general, a supervised algorithm would have to be trained for a certain task and not for a certain task in a certain domain. Ideally an algorithm would be trained for a certain task on data from an arbitrary domain $D_A$ and then also perform well in the same task on data from domain $D_B$, since the creation of training data typically requires a manual annotation effort. In the opinion mining context, the question of domain portability and the reduction of re-labeling efforts has been prominently investigated, with respect to the identification of opinion expressions and the analysis of their polarity (sentiment analysis), in previous research (Aue and Gamon, 2005; Blitzer et al., 2007). Terms as "unpredictable" can express a positive opinion when uttered about the storyline of a movie, but a negative opinion when the handling of a car is described. Hence the effects of training and testing a machine learning algorithm for sentiment analysis on data from different domains have been analyzed in previous research. However, to the best of our knowledge, these effects have not been investigated regarding the extraction of opinion targets.

The contribution of this chapter is a CRF-based approach for opinion target extraction which tackles the problem of domain portability. We first evaluate our approach in four different domains against a state-of-the art supervised baseline system and then evaluate the performance of both systems in a cross-domain setting. We show that the CRF-based approach outperforms the baseline in both settings. Furthermore, we analyze how the different combinations of features we introduce influence the results of our CRF-based approach.

## 4.1 Supervised Approaches to Opinion Target Extraction

Zhuang et al. (2006) present a supervised algorithm for the extraction of opinion expression - opinion target pairs. Their algorithm learns the opinion target candidates and a combination of dependency and part-of-speech paths connecting such pairs from an annotated dataset. They evaluate their system in a cross-validation setup on a dataset of user-generated movie reviews and compare it to the results of the system by Hu and Liu (2004a) as a baseline. Thereby, the system by Zhuang et al. (2006) yields an F-Measure of 0.529 and outperforms the baseline which yields an F-Measure of 0.488 in the task of extracting opinion target - opinion expression pairs.

Kessler and Nicolov (2009) solely focus on identifying which opinion expression is linked to which opinion target in a sentence. They present a dataset of car and camera reviews in which opinion expressions and opinion targets were manually annotated. In (Kessler et al., 2010), they provide a more detailed description of their annotation guidelines, in which opinion expressions (sentiment expressions) are defined as follows: "Sentiment expressions are single or multiword phrases that evaluate an entity. They are linked to the mention they modify through the 'target' relation". Starting with this information, they train a Support Vector Machine-based classifier for identifying related opinion expressions and targets. Their algorithm receives the opinion expression and opinion target annotations as input during runtime. The classifier is evaluated using the algorithm by Bloom et al. (2007) (see Section 3.1 Page 16) as a baseline. The SVM-based approach by Kessler and Nicolov (2009) yields an F-Measure of 0.698, outperforming the baseline which yields an F-Measure of 0.445.

Semi-supervised clustering algorithms have also been employed for opinion target extraction in previous research. Lu and Zhai (2008) introduce an extension to the PLSA algorithm (Hofmann, 1999), which is a statistical approach for the analysis of term co-occurrences in corpora. The concept of this algorithm is that a document consists of a variable number of *topics*, which are in turn represented by the words occurring in the document. The PLSA algorithm identifies these topics autonomously and clusters the words in the corpus with respect to these topics. With the extension by Lu and Zhai (2008) it is possible to control the topics around which the PLSA algorithm shall create the word cluster. These topic concepts have to be provided manually, which results in a semi-supervised algorithm. In (Lu and Zhai, 2008) the authors can however only perform an extrinsic evaluation of their algorithm, since the dataset they work with is not annotated regarding individual opinion expressions or opinion targets.

The challenge of domain portability of a model learned by a supervised algorithm (also referred to as "domain adaptation" (Daumé III and Marcu, 2006; Jiang and Zhai, 2007)) has also been studied in previous research: Aue and Gamon (2005) have investigated this challenge very early in the task of document level sentiment classification (positive / negative). They observe that increasing the amount of training data raises the classification accuracy, but only if the training data is from one source domain. Increasing the training data by mixing domains does not yield any

consistent improvements. Blitzer et al. (2007) introduce an extension to a structural correspondence learning algorithm, which was specifically designed to address the task of domain adaptation. Their enhancement aims at identifying pivot features, which are stable across domains. In a series of experiments in document level sentiment classification they show that their extension outperforms the original structural correspondence learning approach. In their error analysis, the authors observe that the best results were reached when the training - test combinations were *Books - DVDs* or *Electronics - Kitchen appliances*. They conclude that the topical relatedness of the domains is an important factor. Furthermore, they observe that training the algorithm on a smaller amount of data from a similar domain is more effective than increasing the amount of training data by mixing domains.

Choi et al. (2005) focus on the extraction of an aspect also covered by the work of Kim and Hovy (2006), namely the extraction of opinion holders. They combine a supervised information extraction algorithm with a CRF-based approach, which they evaluate on a dataset of newswire. They also aim at identifying opinion expressions related to the holders with their approach and report an inaccurate identification of opinion expressions as a major source of errors in the error analysis. Furthermore, they report that errors in the NLP preprocessing had a considerable negative impact on the results, as well as long and complex sentences which they could not capture well with their features.

### 4.1.1 Baseline System

In the task of opinion target extraction, the supervised algorithm by Zhuang et al. (2006) represents the state-of-the-art on the "movies" dataset described in Chapter 2. Since the "web-services" and "cars" and "cameras" datasets have been released very recently, no algorithm for opinion target extraction has been evaluated on them outside of this thesis yet. To the best of our knowledge, Kessler and Nicolov (2009) were the only ones who employed their dataset for evaluation purposes at the time of this writing. Their algorithm aims at identifying opinion expressions and opinion targets which are related in a sentence. However, their goal is not to automatically identify opinion expressions or opinion targets. Instead, they start from the annotated opinion expressions and target from the gold standard and aim to identify which opinion expression relates to which target in a given sentence.

As it represents the state-of-the-art in supervised opinion target extraction, we employ the algorithm by Zhuang et al. (2006) as a baseline for the evaluation of our CRF-based algorithm. The algorithm learns the following information from the labeled training data:

1. A set of opinion target candidates
2. A set of opinion expression candidates
3. A set of paths in a dependency tree which identify valid opinion target - opinion expression pairs

In the first step, the frequency counts of the opinion target types and the opinion expressions are extracted from the training data. Opinion targets occurring with of

frequency of less than 1% of the overall target frequencies are then discarded. Zhuang et al. (2006) claim, that with the remaining target candidates they can still cover more than 90% of all opinion targets in the test set. We could not reproduce these results: After pruning the target candidates in the described manner, the remaining ones could only cover up to 46% of all targets in the test set (see Section 3.2.3). Furthermore, Zhuang et al. (2006) crawl a set of actors and directors names from the website they collected the movie reviews from. With this list, the algorithm is supposed to identify new opinion targets not occurring in the training data. Since we could not recreate the list of named entities, we decided to modify the algorithm by not discarding any opinion target candidates learned from the training data. As the coverage of the target candidates determines the upper bound regarding recall during the extraction, we expect that this modification is beneficial for the overall performance.

In the second step, the algorithm learns a set of opinion expression candidates. This step is a hybrid of a statistical approach with a bootstrapping extension. First, the 100 most frequently occurring positive and negative opinion expressions are learned from the training data. Then, the algorithm performs a crawling on WordNet (Fellbaum, 1998). For every noun, it checks whether the glosses of the first two synsets contain an opinion expression from the previously created list. If this is the case, then the noun is added to the respective list of opinion expressions. Zhuang et al. (2006) do not precisely state how they proceed if an opinion expression from both the list of positives and the list of negatives is found.

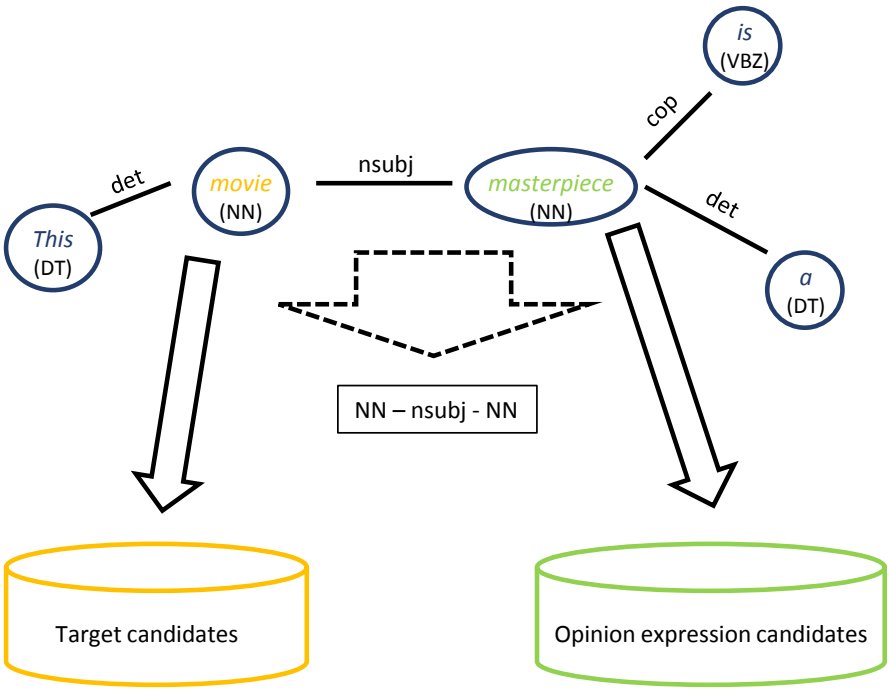
In the third step, the sentences from the training data are parsed and a graph is created which contains the words of the sentence with their respective part-of-speech tag. The nodes are connected with edges, if a dependency relation between them is identified by the parser. The edges are labeled with the corresponding dependency type. For each opinion target - opinion expression pair from the gold standard, the algorithm then extracts the shortest path connecting them in the dependency graph. A path consists of the part-of-speech tags of the nodes and the dependency types of the edges. Example 4.1 shows a typical dependency path.

(4.1) NN - nsubj - NP - amod - JJ

During runtime, the algorithm identifies opinion targets and opinion expressions from the respective candidate lists learned from the training data. The sentences are then parsed, and if a valid path between a target and an opinion expression is found in the list of possible paths, then the pair is extracted. As we are only interested in the performance of the opinion target extraction in this study, during our experiments we employ the opinion expressions from the gold standard and leave out step two. The architecture of the baseline system is shown in Figure 4.1.

The dependency paths only identify pairs of single word targets and opinion expression candidates. Zhuang et al. (2006) do not state how they deal with multiword opinion expression or multiword opinion target candidates. As we are also interested in extracting multiword targets, we extend the algorithm by employing a merging step: Extracted target candidates are merged into a multiword target if they are adjacent in a sentence. Thereby, the baseline system is also capable of extracting multiword opinion targets.

Figure 4.1: Architecture of Zhuang et al. (2006) Baseline




In our experiments, we learn the full set of opinion targets from the labeled training data in the first step. This is slightly different from step one of the approach in (Zhuang et al., 2006), but we expect that this modification is beneficial for the overall performance in terms of recall, as we do not remove any learned opinion targets from the candidate list.

### 4.1.2 CRF-based Approach

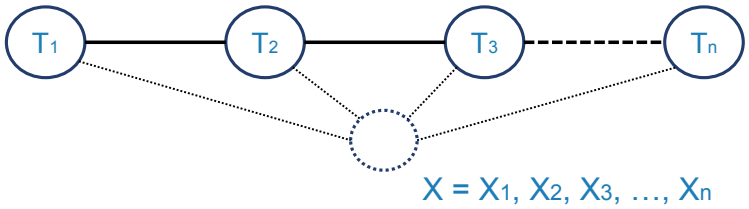
As mentioned in Chapter 1, the task of extracting opinion targets from a given sentence can be viewed as an information extraction problem. In our CRF-based approach, we will model the opinion target extraction as a sequence labeling task, which is frequently encountered in NLP e.g. in part-of-speech tagging or named entity recognition. For many of these tasks, the state-of-the-art algorithms are machine learning based. Especially conditional random fields (Lafferty et al., 2001) have been successfully employed for e.g. the tasks of information extraction (Pinto et al., 2003), part-of-speech tagging and parsing (Sha and Pereira, 2003), or named entity recognition (McCallum and Li, 2003).

A very common approach for performing sequence labeling and segmentation tasks are hidden Markov Models (Rabiner and Juang, 1986), which identify the most probable sequence of labels e.g. for tokens in a sentence. Such hidden Markov Models are generative models, which define a joint probability distribution $p(X, Y)$ where $X$ and $Y$ are variables, e.g. tokens and the corresponding part-of-speech tags. In order

to define the joint probability distribution, a generative model would have to iterate over all possible observation sequences, i.e. sequences of token / part-of-speech tag combinations in a sentence. For most tasks, this enumeration is impossible due to the amount of possible combinations. In order to obtain a computable and trainable model, a typical approach is to hence represent the elements as isolated units, which are independent of each other. This assumption may hold for a few tasks or domains, but it typically does not adequately represent the problem. Conditional random fields (CRF) address this drawback by defining a conditional probability $p(Y|x)$ over label sequences given a certain observation sequence $x$. Conditional models are then employed to label an unknown observation sequence $x_*$ by selecting the label sequence $y_*$ which maximizes the conditional probability $p(y_*|x_*)$. Thereby, the model does not require any independence assumptions.

CRFs are probabilistic frameworks for the task of labeling and segmenting sequential data. They can be represented as undirected graphs in which the vertices represent the variables, e.g. tokens in a sentence. A typical approach in NLP related tasks is to model the variables as a chain, with adjacent variables (=tokens in a sentence) being connected by an edge. Our goal is to extract individual instances of opinion targets from sentences which contain an opinion expression. This can be modeled as a sequence segmentation and labeling task. Figure 4.2 shows our representation of the sentences as a linear chain.

Figure 4.2: CRF Graph Representation of Sentences




The CRF algorithm receives a sequence of tokens $T_1...T_n$ for which it has to predict a sequence of labels $l_1...l_n$. We represent the possible labels following the IOB scheme: *B-Target*, identifying the beginning of an opinion target, *I-Target* identifying the continuation of a target, and *O* for other (non-target) tokens. We model the sentences as a linear chain CRF, which is based on an undirected graph. In the graph, each node corresponds to a token in the sentence and edges connect the adjacent tokens as they appear in the sentence. In our experiments, we use the CRF implementation from the Mallet toolkit[1].

In the following, we will describe the features we employ as input for our CRF-based approach. As the development data, we used 29 documents from the movies dataset, 23 documents from the "web-services" dataset and 15 documents from the "cars" & "cameras" datasets. The development to training+test data ratios for each dataset are shown in Table 4.1.

[1] `http://mallet.cs.umass.edu/`

Table 4.1: Ratios of Development to Training+Test Data (in Documents)

| Dataset | Development | Training+Test |
|---|---|---|
| movies | 1.5% | 98.5% |
| web-services | 9.8% | 90.2% |
| cars | 4.4% | 95.6% |
| cameras | 8.3% | 91.7% |

**Token**

This feature represents the string of the current token as a feature. Even though this feature is rather obvious, it can have considerable impact on the target extraction performance. If the vocabulary of targets is rather compact for a certain domain (corresponding to a low target type / target ratio), the training data is likely to contain the majority of the target types. The token string should hence be a good indicator. We will refer to this feature as **tk** in our result tables.

**POS**

This feature represents the part-of-speech tag of the current token as identified by the Stanford POS Tagger[2]. It can provide some means of lexical disambiguation, e.g. indicate that the token "sounds" is a noun and not a verb in a certain context. At the same time, the CRF algorithm is provided with additional information to extract opinion targets which are multiword expressions, i.e. noun combinations. We will refer to this feature as **pos** in our result tables.

**Direct Dependency Link**

Previous research has successfully employed paths in the dependency parse tree to link opinion expressions and the corresponding targets (Zhuang et al., 2006; Kessler and Nicolov, 2009). Both works identify direct dependency relations such as "amod" and "nsubj" as the most frequent and at the same time highly accurate connections between a target and an opinion expression. An example for the "nsubj" relation can be found in the sentence: "This movie is a masterpiece", in which it connects the opinion target "movie" and the opinion expression "masterpiece". We hence label all tokens which have a direct dependency relation to an opinion expression in a sentence. The Stanford Parser[3] is employed for the constituent and dependency parsing. We will refer to this feature as **dLn** in our result tables.

**Nearest Noun Phrase**

From the work of Zhuang et al. (2006) we can infer that opinion expressions and their target(s) are not always connected via short paths in the dependency parse tree. The third most frequent dependency - part-of-speech path which they learn during

[2] `http://nlp.stanford.edu/software/tagger.shtml`

[3] `http://nlp.stanford.edu/software/lex-parser.shtml`

their training phase is: "NN - nsubj - VB - dobj - NN", in which the opinion expression and the opinion target are connected via two dependency relations. Kessler and Nicolov (2009) even report that they discovered 1002 unique dependency - part-of-speech-paths connecting opinion expressions and opinion targets on their dataset. Since we cannot capture such complex paths with the above mentioned feature, we introduce another feature which acts as heuristic for identifying the target to a given opinion expression. Hu and Liu (2004a) and Yi et al. (2003) have shown that (base) noun phrases are good candidates for opinion targets in the datasets of product reviews. Our experiments in Section 3.2.2 have shown, that the "nearest noun phrase heuristic" performs quite reasonably in identifying opinion targets for given opinion expressions. We therefore label the token(s) in the closest noun phrase regarding word distance to each opinion expression in a sentence. We will refer to this feature as **nNp** in our result tables.

**Opinion Sentence**

With this feature, we simply label all tokens occurring in a sentence containing an opinion expression. This feature shall enable the CRF algorithm to distinguish between the occurrence of a certain token in a sentence which contains an opinion vs. a sentence without an opinion. We will refer to this feature as **sSn** in our result tables.

## 4.2 Experiments and Results

In our experiments, we again employ the datasets we presented in Chapter 2 and the evaluation metrics as presented for the opinion target extraction task in 3.2.1. We investigate the performance of the baseline and the CRF-based approach for opinion target extraction in a single- and cross-domain setting. The single-domain setting assumes that there is a set of training data available for the same domain as the domain the algorithm is being tested on. In this setup, we will both run the baseline and our CRF-based system in a 10-fold cross-validation and report results macro-averaged over all runs.[4] In the cross-domain setting, we will investigate how the algorithm performs if given training data from domain $A$ while being tested on another domain $B$. In this setting, we will train the algorithm on the entire dataset $A$, and test it on the entire dataset $B$. We hence report one micro-averaged result set.[5] In Section 4.2.1, we present the results of both the baseline system and our CRF-based approach in the single-domain setting. In Section 4.2.2, we present the results of the two systems in the cross-domain opinion target extraction.

Table 4.2: Single-Domain Opinion Target Extraction with Zhuang Baseline

| Dataset | Precision | Recall | F-Measure |
|---|---|---|---|
| movies | 0.663 | 0.592 | 0.625 |
| web-services | 0.624 | 0.394 | 0.483 |
| cars | 0.259 | 0.426 | 0.322 |
| cameras | 0.423 | 0.431 | 0.426 |

### 4.2.1 Single-Domain Results

#### Zhuang Baseline

As shown in Table 4.2, the state-of-the-art algorithm of Zhuang et al. (2006) performs best on the "movies" dataset and worst on the "cars" dataset. The results on the "movies" dataset are higher than originally reported in (Zhuang et al., 2006) (precision 0.483, recall 0.585, F-Measure 0.529). We assume that this is due to two reasons:

1. In our task, the algorithm uses the opinion expression annotations from the gold standard.

2. We do not remove any learned opinion target candidates from the training data (see Section 4.1.1).

During training, we observed that for each dataset the lists of possible dependency paths (see Example 4.1) contained several hundred entries, many of them only occurring once. We assume that the recall of the algorithm is limited by a large variety of possible dependency paths between opinion targets and opinion expressions, since the algorithm cannot link targets and opinion expressions in the test set if there is no valid candidate dependency path. Furthermore, we observe that for the "cars" dataset the size of the dependency path candidate list (6642 entries) is approximately five times larger than the dependency graph candidate list for the "web-services" dataset (1237 entries). At the same time, the list of target candidates of the "cars" dataset is approximately eight times larger than the target candidate list for the web-services dataset. We assume that a large number of both the target candidates as well as the dependency path candidates introduce many false positives during the target extraction, hence lowering the precision of the algorithm on the "cars" dataset considerably.

[4]By macro-averaging we mean, that we calculate one F-Measure result per fold, and then report the average of these 10 folds. This approach is possible because our folds are of equal size.

[5]In this setting by calculating precision, recall and F-Measure across all instances and documents in the evaluation dataset.

Table 4.3: Single-Domain Opinion Target Extraction with our CRF-based Approach

| | **movies** | | | **web-services** | | | **cars** | | | **cameras** | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Features** | **Prec** | **Rec** | **F-M** | **Prec** | **Rec** | **F-M** | **Prec** | **Rec** | **F-M** | **Prec** | **Rec** | **F-M** |
| tk, pos | 0.639 | 0.133 | 0.220 | 0.500 | 0.051 | 0.093 | 0.438 | 0.110 | 0.175 | 0.300 | 0.085 | 0.127 |
| tk, pos, nNp | 0.542 | 0.181 | 0.271 | 0.451 | 0.272 | 0.339 | 0.570 | 0.354 | 0.436 | 0.549 | 0.375 | 0.446 |
| tk, pos, dLn | 0.777 | 0.481 | 0.595 | 0.634 | 0.380 | 0.475 | 0.603 | 0.372 | 0.460 | 0.569 | 0.376 | 0.453 |
| tk, pos, sSn | 0.673 | 0.637 | 0.653 | 0.604 | 0.397 | 0.476 | 0.453 | 0.180 | 0.257 | 0.398 | 0.172 | 0.238 |
| tk, pos, dLn, nNp | **0.792** | 0.481 | 0.598 | 0.620 | 0.354 | 0.450 | 0.603 | 0.389 | 0.473 | 0.596 | 0.425 | 0.496 |
| tk, pos, sSn, nNp | 0.662 | 0.656 | 0.659 | 0.664 | 0.461 | 0.544 | 0.564 | 0.370 | 0.446 | 0.544 | 0.381 | 0.447 |
| tk, pos, sSn, dLn | 0.791 | 0.477 | 0.594 | 0.654 | 0.501 | 0.568 | 0.598 | 0.384 | 0.467 | 0.586 | 0.391 | 0.468 |
| tk, pos, sSn, dLn, nNp | 0.749 | **0.661** | **0.702** | **0.722** | **0.526** | **0.609** | **0.622** | **0.414** | **0.497** | 0.614 | **0.423** | **0.500** |
| pos, sSn, dLn, nNp | 0.672 | 0.441 | 0.532 | 0.612 | 0.322 | 0.422 | 0.612 | 0.369 | 0.460 | **0.674** | 0.398 | 0.500 |

### CRF-based Approach

Table 4.3 shows the results of the opinion target extraction using the CRF algorithm. Row 8 contains the results of the feature configuration, which yields the best performance regarding F-Measure across all datasets. We observe that our approach significantly outperforms the Zhuang et al. (2006) baseline on all datasets.[6] The gain in F-Measure is between 0.077 in the movies domain and 0.175 in the cars domain. Although the CRF-based approach clearly outperforms the baseline system on all four datasets, we also observe the same general trend regarding the individual results: The CRF yields the best results on the "movies" dataset and the worst results on the "cars" dataset. This trend in the results is identical to the outcome of our experiments in Chapter 3. The supervised approaches yield the best results on the dataset which has the smallest lexical variability regarding opinion targets (the "movies" dataset) and the worst results on the dataset which has the highest lexical variability regarding opinion targets ("cars" dataset).

As shown in the first row of Table 4.3, the results when using just the token string and part-of-speech tags as features are very low, especially regarding recall. If we add the feature based on the nearest noun phrase heuristic (row 2), the recall is improved on all datasets, while the precision is slightly lowered on the "movies" and "web-services" datasets. The dependency path based feature performs better compared to the nearest noun phrase heuristic as shown in row 3. The precision is considerably increased on all datasets and at the same time, we observe an increase of recall on all datasets. The observation made in previous research that short paths in the dependency graph are a high precision indicator of related opinion expressions - opinion targets (Kessler and Nicolov, 2009) is confirmed on all datasets. Adding the information regarding opinion sentences to the basic features of the token string and the part-of-speech tag (row 4) yields the biggest improvements regarding F-Measure on the "movies" and "web-services" datasets (+0.433 / +0.383). On the "cars" and the "cameras" datasets, the recall is relatively low. We assume that this is due to the high lexical variability regarding opinion targets: If there are many targets which only occur once, then the probability is higher, that some of them do not occur in the training data. The CRF algorithm then encounters many actual opinion targets in the test set, which have not occurred in the training data and will hence not be extracted.

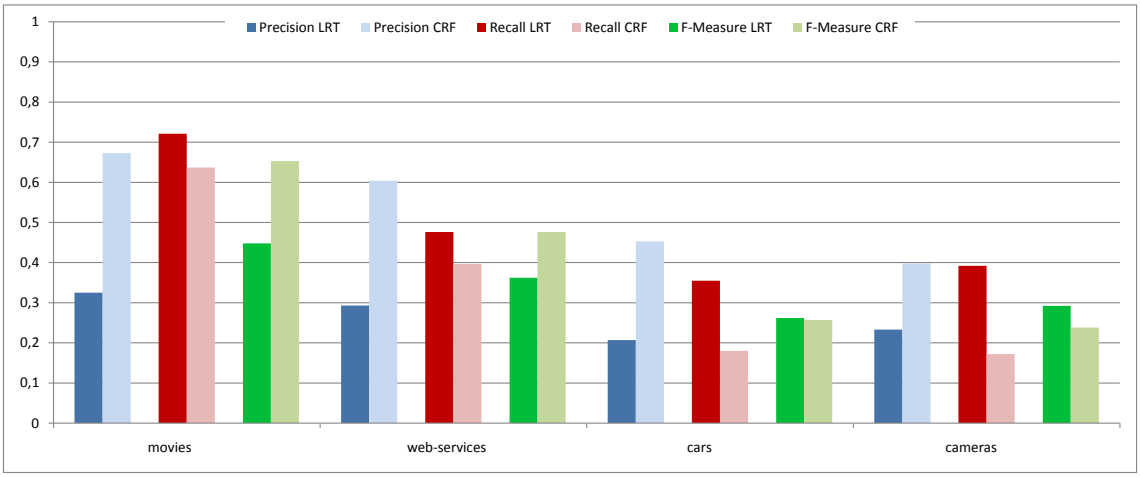
As shown in row 5 of Table 4.3, if we combine the dependency graph based feature with the nearest noun phrase heuristic, the results regarding F-Measure are consistently higher than the results of these features in isolation (rows 2 - 4) on all datasets. We conclude that these two features are complementary, as they apparently indicate different kinds of opinion targets which are then correctly extracted by the CRF. If we combine each of the opinion expression related features with the label which identifies opinion sentences in general (rows 6 & 7), we observe that this feature is also complementary to the others. On all datasets, the results regarding F-Measure are consistently higher compared to the features in isolation (rows 2 - 4). Row 8 shows the results of all features in combination. Again, we observe the complementarity of the features, as the results of this feature combination are the best regarding F-Measure across all datasets.

[6]Significance of improvements is tested using a paired two-tailed t-test with $p \leq 0.05$.

In row 9 of Table 4.3, we exclude the token string as a feature. In comparison to the full feature combination of row 8, we observe a significant decrease of F-Measure on the movies and the web-services dataset. On the "cars" dataset, we only observe a slight decrease of recall. Interestingly on the "cameras" dataset, we even observe a slight increase of precision which compensates a slight decrease of recall, in turn resulting in a stable F-Measure of 0.500 as in the full feature set of row 8.

In the following, we will compare the performance of the unsupervised algorithms from Chapter 3 with our CRF-based approach and if possible the supervised baseline. We will structure the analysis along the four Settings introduced in Section 3.2.2.

Figure 4.3: Single-Domain Opinion Target Extraction with Gold Standard Opinion Sentences: LRT vs. CRF



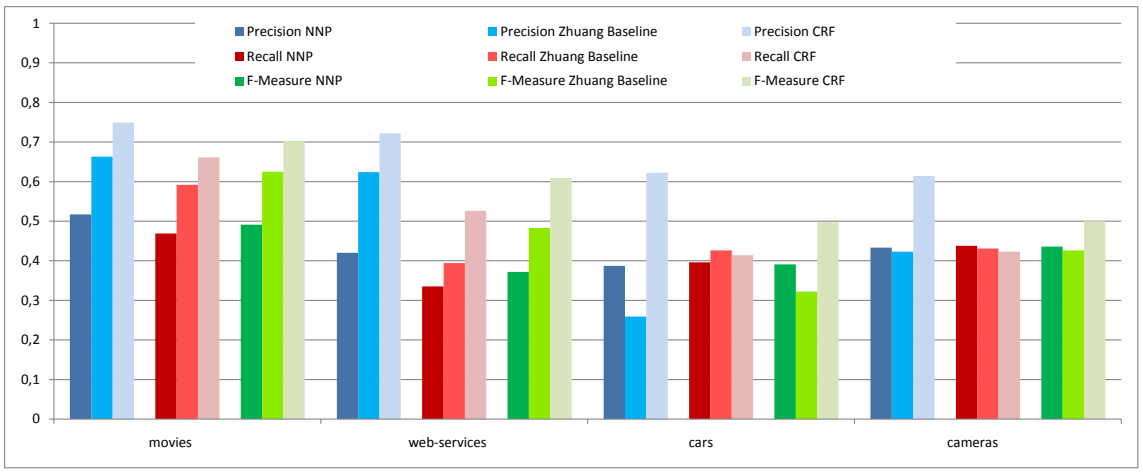

**Setting I: Gold Standard Opinion Sentences**

As described in Section 4.1.1, the baseline system learns dependency - part-of-speech paths connecting opinion expressions and opinion targets from the training data. However, in this experimental setting we only employ opinion bearing sentences, in which individual opinion expressions have not been identified. We can therefore not employ the baseline system in our experiments, since it relies on the identification of individual opinion expressions in order to perform the training phase. We will instead compare the results of the best performing unsupervised algorithm from Chapter 3 with the CRF configurations, which use the *token, part-of-speech tags* and *opinion sentence* features. We have created an overview of the results in Figure 4.3. As shown in the Table, our CRF-based approach clearly outperforms the LRT regarding precision on all four datasets. But at the same time, the recall of the CRF approach is consistently lower compared to the LRT. On the "movies" and the "web-services" datasets, the CRF can again yield a competitive recall resulting in the higher F-Measure. However on the "cars" and "cameras" datasets, which are very challenging to both the unsupervised and supervised algorithms due to the high lexical variability, the LRT yields a slightly higher F-Measure. However, in a scenario in which one has to extract opinion targets while only opinion bearing sentences are identified, we believe that it is still sensible to invest into

the labeling effort in order to generate training data for a supervised approach for two reasons: If the lexical variability regarding opinion targets in a dataset is low, then the supervised CRF-based approach can clearly outperform the unsupervised LRT-based approach. If the lexical variability is high, then both algorithms will probably only yield relatively low results, with the supervised CRF-based approach either being on par with the unsupervised approach or slightly worse.

We assume, that the low results of the CRF-based approach are also due to our experimental setup. The algorithm is trained on the entire dataset, which means both the opinion and the non-opinion sentences. In this setup and feature setting, the CRF will encounter many contradicting indicators regarding the opinion targets. A certain term, e.g. "zoom" will probably occur as an opinion target $x$ times, but also as a non-target $y$ times. Since the CRF does not have any additional information (more features), it will not be able to make a clear decision whether a certain term is a target in a given context (sentence) or not. Better results might be achieved with a different setup: By filtering the training data for only the opinion sentences, the algorithm will probably receive more consistent indicators regarding the target terms. The documents which are to be classified could in turn also be pre-filtered for the opinion sentences, hence reducing the candidates of false positives. Such an approach might be a promising direction for future work.

Figure 4.4: Single-Domain Opinion Target Extraction with Gold Standard Opinion Expressions: NNP vs. Baseline vs. CRF

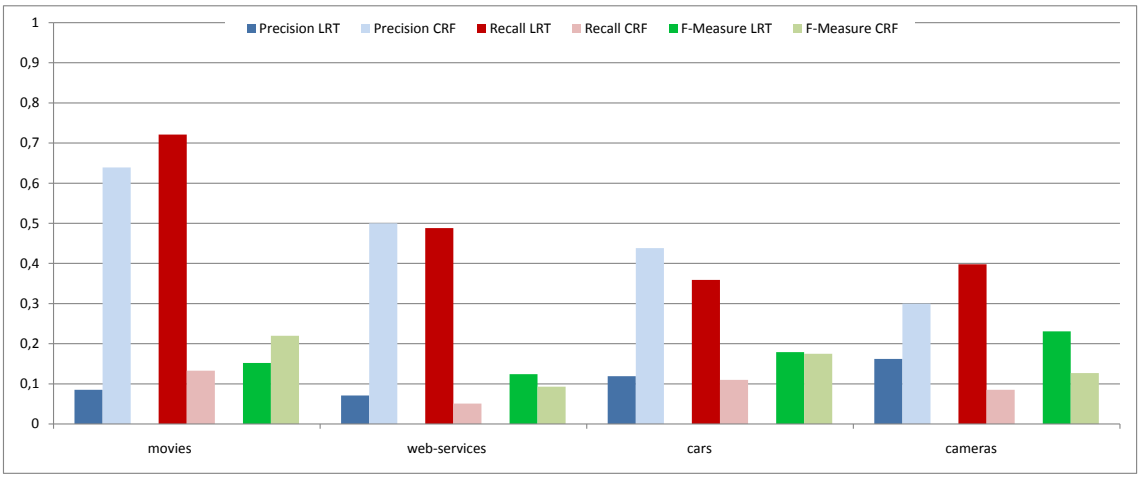



**Setting II: Gold Standard Opinion Expressions**

Figure 4.4 shows a comparative overview of the two supervised approaches and the nearest noun phrase heuristic (NNP) as the best performing unsupervised approach from Chapter 3. On the "movies" and the "web-services" datasets, even the Zhuang et al. (2006) baseline clearly outperforms the *nearest noun phrase* heuristic. However, on the "cars" dataset the unsupervised *nearest noun phrase* heuristic yields a considerably higher precision regarding the opinion target extraction than the Zhuang et al. (2006) baseline, which in turn leads to a higher F-Measure in the opinion target extraction task. The precision of the baseline is as low as the results

yielded by the Likelihood Ratio Test based approaches on this dataset. Apparently, even the learned dependency paths do not prevent the baseline from extracting a considerable number of false positives on this dataset. As mentioned above, the dependency path candidate list learned by the baseline is very large for the cameras dataset (6642 entries). In fact, it contains the most entries by far, the second largest list is learned on the "movies" dataset with 4428 entries. The high variation in the dependency path candidate list suggests that the opinion sentences are very diversified in the "cars" dataset, which is however not reflected by the average sentence length (see Table 2.1). Our CRF-based approach yields a significantly higher precision than the *nearest noun phrase* heuristic on this dataset and still reaches the highest F-Measure. On the "cameras" dataset, the *nearest noun phrase* heuristic yields slightly higher precision and recall than the unsupervised baseline. With all of our features employed, the CRF-based approach yields considerably higher precision while having slightly lower recall, which in turn still results in the highest F-Measure. Our features which combine information about the distance to the opinion expression in a sentence and information about the dependency relations enable the CRF algorithm to extract the opinion targets at higher precision than the unsupervised NNP approach, while maintaining an at least equally high recall.

Figure 4.5: Single-Domain Target Extraction without Opinion Identification: LRT vs. CRF




**Setting III: No Opinion Identification**

In this Setting, we can again not include the Zhuang et al. (2006) baseline system, as it relies on the identification of opinion expressions for the extraction of opinion targets. As shown in Figure 4.5, the LRT and the CRF-based approach perform very differently without any opinion identification. The LRT-based approach typically has high recall and low precision, since every occurrence of a highly ranked target candidate will be extracted. The CRF-based approach on the other hand yields a low recall and high precision. This is due to the fact that it receives contradicting information about whether a certain target (string) is a target or not. In some sentences, a given term is a target and in others it is not, but the CRF is not able

to detect any pattern on which this decision can be based. Hence it only learns and extracts a few terms which (coincidentally) exclusively occur as targets in the training data. This behavior leads to the higher precision, but low recall. This is identical to the problem observed in Setting I, but here we have no chance of pre- or post-filtering the training and test set in terms of classifying sentences as opinion-bearing or not.

**Setting IV: MPQA Opinion Expressions**

Analogous to our experiments in Section 3.2.2, Setting IV, we have run some additional experiments in which we did not rely on the gold standard opinion expressions, but employed the MPQA lexicon for opinion expression identification. We again used the "strong" subjectivity clues from the MPQA lexicon and ran the algorithm with the best performing full feature set *tk, pos, sSn, dLn, nNp* from the previous experiments. The results of these experiments are shown in Table 4.4.

Table 4.4: Single-Domain Opinion Target Extraction with CRF and MPQA Opinion Expressions

| Dataset | Precision | Recall | F-Measure |
|---|---|---|---|
| movies | 0.565 | 0.214 | 0.309 |
| web-services | 0.412 | 0.167 | 0.236 |
| cars | 0.441 | 0.124 | 0.192 |
| cameras | 0.357 | 0.138 | 0.198 |

As evident from the results, the opinion target extraction decreases considerably regarding F-Measure on all datasets. This is due to very low recall values. If we compare these results to the configuration in Table 4.3 (Page 68) which just employs tokens and their part-of-speech tags as features, we observe that on the "movies" and the "web-services" datasets the results regarding F-Measure are higher than when the MPQA lexicon is employed. As shown in the error analysis in Section 3.2.3, both precision and recall of the opinion expression identification with the MPQA lexicon are very low. The three features which are based on the opinion expressions apparently even disturb the CRF from learning simply the target strings as a reliable indicator. Hence we conclude that unless the performance of the opinion expression identification can be improved, it is better to solely rely on the target strings (and part-of-speech tags) as features for the target extraction. State-of-the-art algorithms for the identification of opinion expressions, which are based on machine learning algorithms and far more complex than the lexicon based-approach which we employ, do however yield considerably better results than the ones we report in Table 4.4 (Breck et al., 2007; Johansson and Moschitti, 2010). The best configuration by Johansson and Moschitti (2010) reaches a precision of 0.616 and a recall of 0.547 leading to an F-Measure of 0.579 in the extraction of opinion expressions when the same evaluation strategy which we employ is used.[7]

[7]Exact boundary matches are required between the extracted opinion expressions and the opinion expressions in the gold standard.

**Error Analysis**

We performed a quantitative error analysis on the results of the best-performing CRF-based approach in the single-domain setting. We analyzed the results of all ten folds of the respective datasets. In doing so, we focussed on misclassifications of B-Target and I-Target instances, as the recall is consistently lower than the precision across all datasets. We observe that most of the recall errors result from one-word opinion targets or the beginning of opinion targets (B-Targets) being misclassified as non-targets (movies 83%, web-services 73%, cars 68%, cameras 64% of all recall errors). For the majority of these misclassifications neither the *short dependency path* nor the *nearest noun phrase* features were set / enabled (movies 82%, web-services 56%, cars 64%, cameras 61% of all recall errors). This means, that for many actual opinion targets which were not extracted there was neither a short dependency path to the opinion expression in the sentence, nor was it the nearest noun phrase with respect to the opinion expression. An example of such a sentence, in this case from a camera review, in shown in Example 4.2.

(4.2) A lens cap and a strap may not sound very important, but it [sic] makes a *huge difference* in the speed and usability of the camera.

In this sentence, the *dLn* and *nNp* features both labeled "speed" which was incorrectly extracted as the target of the opinion. None of the actual targets "lens cap", "strap" and "camera" have a short dependency path to the opinion expression and "speed" is simply the closest noun (phrase) to it. Note that although both "speed" and "usability" are attributes of a camera, the opinion in this sentence is about the "lens cap" and "strap", hence only these attributes are annotated as targets. Our results indicate that the *dependency link* and *nearest noun phrase* features are complementary, but our error analysis indicates there are quite a few cases in which the opinion target is neither directly related to the opinion expression in the dependency graph nor close to it in the sentence. As shown in the example above, anaphoric expression are also prominently used to refer to the opinion target, even within a sentence.

### 4.2.2 Cross-Domain Results

**Zhuang Baseline**

Table 4.5 shows the results of the baseline system by Zhuang et al. (2006) in the cross-domain setting. The best results regarding precision, recall and F-Measure are highlighted by boldface for the respective datasets. We observe that the results on all domain combinations are very low. A quantitative error analysis has revealed that there is hardly any overlap in the opinion target candidates between domains, as reflected by the low recall in all configurations. The vocabularies of the opinion targets are too different, hence the performance of the algorithm by Zhuang et al. (2006) is so low. The overlap regarding the learned dependency - part-of-speech paths between different domains was however higher: Especially identical short paths could be found across domains which at the same time typically occurred quite often. Examples of such short paths which frequently occur across domains are shown in the following:

Table 4.5: Cross-Domain Opinion Target Extraction with Zhuang et al. (2006) Baseline

| Training | Test | Precision | Recall | F-Measure |
|---|---|---|---|---|
| web-services | movies | **0.194** | 0.032 | 0.055 |
| cars | movies | 0.032 | 0.034 | 0.033 |
| cameras | movies | 0.155 | 0.084 | **0.109** |
| cars + cameras | movies | 0.071 | **0.104** | 0.084 |
| web-services + cars + cameras | movies | 0.070 | 0.103 | 0.083 |
| movies | web-services | **0.311** | 0.073 | **0.118** |
| cars | web-services | 0.086 | 0.091 | 0.089 |
| cameras | web-services | 0.164 | 0.081 | 0.108 |
| cars + cameras | web-services | 0.086 | **0.104** | 0.094 |
| movies + cars + cameras | web-services | 0.074 | 0.100 | 0.080 |
| movies | cars | 0.182 | 0.014 | 0.026 |
| web-services | cars | 0.218 | 0.028 | 0.049 |
| cameras | cars | **0.250** | 0.121 | 0.163 |
| cameras + web-services | cars | 0.247 | **0.131** | **0.171** |
| movies + web-services | cars | 0.246 | 0.045 | 0.076 |
| movies | cameras | 0.108 | 0.012 | 0.022 |
| web-services | cameras | **0.268** | 0.048 | 0.082 |
| cars | cameras | 0.125 | **0.160** | **0.140** |
| cars + web-services | cameras | 0.119 | 0.157 | 0.136 |
| movies + web-services | cameras | 0.245 | 0.063 | 0.100 |

(4.3) nn - NSUBJ - nn

(4.4) nn - AMOD - JJ

For future work, it might be interesting to investigate how the algorithm by Zhuang et al. (2006) performs in the cross-domain setting if the target candidate selection is performed differently, e.g. with an unsupervised approach as discussed in Chapter 3. A possible approach would be to still learn the dependency - part-of-speech paths from data of the training domain $D_{Train}$, but instead of also learning the target candidates from $D_{Train}$, e.g. the Likelihood Ratio Test could be applied to the test domain $D_{Test}$ for the target candidate selection. The opinion target candidates selected by the LRT, which are found in a valid dependency - part-of-speech path to an opinion expression, could then be extracted as the opinion targets.

### CRF-based Approach

The results of the cross-domain opinion target extraction with the CRF-based algorithm are shown in Table 4.6. Due to the increase of the number of system configurations introduced by the training - test data combinations, we had to limit the

Table 4.6: Cross-Domain Extraction with our CRF-based Approach

| | Test | | | | | |
|---|---|---|---|---|---|---|
| | **web-services** | | | **movies** | | |
| **Training** | **Pre** | **Rec** | **F-M** | **Pre** | **Rec** | **F-M** |
| web-services | - | - | - | 0.560 | 0.339 | 0.422 |
| movies | **0.565** | 0.219 | 0.316 | - | - | - |
| cars | 0.538 | 0.248 | 0.340 | 0.642 | 0.382 | 0.479 |
| cameras | 0.529 | 0.256 | 0.345 | 0.642 | 0.408 | 0.499 |
| movies + cars | 0.554 | 0.249 | 0.344 | - | - | - |
| movies + cameras | 0.530 | **0.273** | **0.360** | - | - | - |
| movies + cars + cameras | 0.562 | 0.250 | 0.346 | - | - | - |
| cars + cameras | 0.538 | 0.254 | 0.345 | 0.641 | 0.395 | 0.489 |
| web-services + cars | - | - | - | **0.651** | 0.396 | 0.492 |
| web-services + cameras | - | - | - | 0.642 | **0.435** | **0.518** |
| web-services + cars + cameras | - | - | - | 0.639 | 0.405 | 0.496 |
| | **cars** | | | **cameras** | | |
| | **Pre** | **Rec** | **F-M** | **Pre** | **Rec** | **F-M** |
| web-services | 0.391 | 0.277 | 0.324 | 0.505 | 0.330 | 0.399 |
| movies | 0.512 | 0.307 | 0.384 | 0.550 | 0.303 | 0.391 |
| cars | - | - | - | 0.665 | 0.369 | 0.475 |
| cameras | **0.589** | 0.384 | **0.465** | - | - | - |
| cameras + movies | 0.567 | **0.394** | **0.465** | - | - | - |
| cameras + web-services | 0.572 | 0.381 | 0.457 | - | - | - |
| movies + web-services | 0.489 | 0.327 | 0.392 | 0.553 | 0.339 | 0.421 |
| movies + cars | - | - | - | 0.634 | 0.376 | 0.472 |
| web-services + cars | - | - | - | **0.678** | 0.376 | **0.483** |
| web-services + movies + cars | - | - | - | 0.635 | **0.378** | 0.474 |
| movies + web-services + cameras | 0.549 | 0.381 | 0.450 | - | - | - |

results of the feature combinations reported in the Table. The feature combination *pos, sSn, nNp, dLn* yields the best results regarding F-Measure, hence we report its result as the basic feature set. Since the results in the single-domain setting were already so low when the MPQA lexicon was employed for the opinion expression identification, we decided not to employ it in our evaluation in the cross-domain target extraction. The best results regarding precision, recall and F-Measure are highlighted in boldface for the respective datasets. When comparing the results of the best performing feature / training data combination of the CRF-based approach with the baseline, we observe that our approach considerably outperforms the baseline on all four domains. The best-performing configuration of the CRF-based approach outperforms the best-performing baseline configuration by 0.409 regarding F-Measure in the movies domain, by 0.242 regarding F-Measure in the web-services domain, by 0.294 regarding F-Measure in the cars domain and by 0.343 regarding

F-Measure in the cameras domain.

**Effects of Features**

Interestingly with the best performing feature combination from the single-domain extraction, the results regarding recall in the cross-domain extraction are very low (e.g. on "web-services" dataset - precision: 0.301, recall: 0.079, F-Measure: 0.125). This is due to the fact that the CRF attributes a relatively large weight to the token string feature. As we also observed in the analysis of the baseline results, the overlap of the opinion target vocabularies between domains is low. The CRF therefore only encounters a few target candidates in the test set which it has also observed in the training data. This results in a very small number of targets being extracted by the CRF. As shown in Table 4.6, by removing the token feature from the CRF configuration, we can reach promising results regarding F-Measure. With the CRF-based approach, it is therefore possible to overcome the differences in the vocabularies between training and test domains.

**Effects of Training Data**

When analyzing the results of the different training - test domain configurations we observe the following: In isolation, the training data from the "cameras" domain consistently yields the best results regarding F-Measure when the algorithm is run on the datasets from the other three domains. This is particularly interesting since the "cameras" dataset is the smallest of the four (see Table 2.1). We investigated whether the CRF algorithm was overfitting to the training datasets by reducing their size to the size of the "cameras" dataset. However, the reduction of the training data sizes never improved the extraction results regarding F-Measure for the "movies", "web-services" and "cars" datasets. The good results when training on the cameras dataset are in line with our observations from Section 4.2.1. We noticed that on the "cameras" dataset the results regarding F-Measure remained stable if the token feature is not used in the training.

In isolation, training only on the "cars" data yields the second highest results on the "movies" and "web-services" datasets and the highest results regarding F-Measure on the "cameras" data. However, the results of the "cars + cameras" training data combination indicate that the "cars" data does not contribute any additional information during the learning, since the results on both the "movies" and the "web-services" datasets are not higher than when training only on the "cameras" data. From our analysis we could not deduct any patterns which indicated properties that a dataset should have in order to be a good candidate for the training data. An option for future work could be to analyze the distribution of the features in the training data in order to identify possible patters which reveal why, when training on only one dataset, the "cameras" dataset yields the best results.

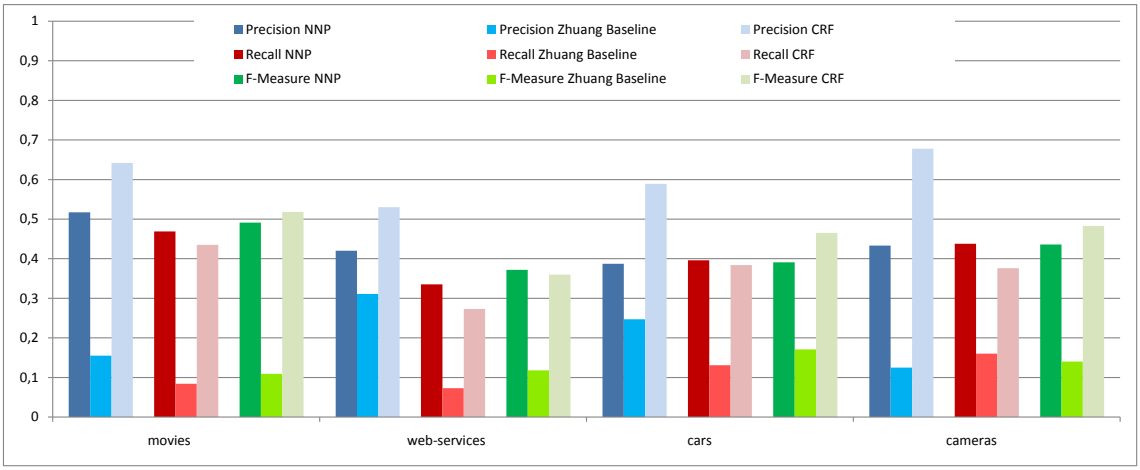
Our results also confirm the insights gained by Blitzer et al. (2007), who observed that in cross-domain polarity analysis adding more training data is not always beneficial. Apparently, even the smallest training dataset ("cameras") contains enough feature instances to learn a model which performs well on the test set.

We observe that the results of the cross-domain extraction regarding F-Measure come relatively close to the results of the single-domain setting, especially if the token string feature is removed there (see Table 4.3 row 9). On the "cars" and

the "cameras" dataset, the cross-domain results are even closer to the single-domain results. The features we employ seem to generalize well across domains and compensate for the difference between the training and test data and the lack of information regarding the target vocabulary.

In Figure 4.6, we compile an overview of the best performing unsupervised approach from Chapter 3, the best configurations of the Zhuang et al. (2006) baseline, and the CRF-based approach. We observe that the *nearest noun phrase* heuristic significantly outperforms the supervised baseline regarding recall on all datasets. On the "web-services" dataset, the unsupervised approach even slightly outperforms the CRF-based approach regarding F-Measure, due to a higher recall. On the other three datasets, the CRF-based approach yields the highest results regarding F-Measure. But our results show that the unsupervised approach based on the simple nearest noun phrase heuristic is quite a competitive baseline. Especially considering that we selected the training dataset combination which yielded the best results for Figure 4.6. In a real-world scenario, the decision regarding the training data could be quite difficult, given that there was no consistent trend towards or against certain combinations in our results.

Figure 4.6: Cross-domain Opinion Target Extraction with Nearest Noun Phrase Heuristic vs. best Zhuang et al. (2006) Baseline Configuration vs. best CRF Configuration




## 4.3 Chapter Summary

In this chapter, we have presented a CRF-based approach for opinion target extraction. We have evaluated it against a state-of-the-art supervised algorithm in a single- and cross-domain setting. We have presented a comparative evaluation of our approach on datasets from four different domains. The CRF-based approach outperforms the baseline on all datasets in the single- and the cross-domain setting. We have furthermore compared the supervised algorithms against the nearest noun phrase heuristic, which is the best performing unsupervised approach from

Chapter 3. Especially in the cross-domain setting, the nearest noun phrase heuristic yields very competitive results, as there is no possible overfitting on the training domain(s). The CRF-based approach outperforms the unsupervised algorithm on three out of the four datasets. We conclude that the features we employ generalize well across domains, also given that the opinion target vocabularies are substantially different. By leaving out the opinion target token strings from the feature set, we can abstract the algorithm from a lexicalized form which learns the opinion target vocabulary of the given training domain.

Three of the five features we employ in the training phase use information regarding the opinion expressions in the sentences. Our evaluation in Chapter 3 has shown that the general domain MPQA lexicon does not yield satisfying results regarding the opinion expression identification. For future work, we might therefore investigate whether machine learning algorithms, which are specifically designed for the problem of domain adaptation (Blitzer et al., 2007; Jiang and Zhai, 2007), can yield better results in a cross-domain opinion expression identification. Another option for future work might be to investigate how the supervised approaches which represent the state-of-the-art in single-domain opinion expression identification (Breck et al., 2007; Johansson and Moschitti, 2010) perform in a cross-domain setting.

# Chapter 5

# Extracting Anaphoric Opinion Targets

In Chapters 3 and 4, we have investigated the extraction of opinion targets on a word / phrase level with supervised and unsupervised methods. A phenomenon which we have not covered so far regards targets which are references by anaphoric expressions. Consider the following example in which the opinion targets are underlined and the opinion expressions are shown in boldface:

(5.1) I think that it is **terrific** how well The Wizard of Oz has held up over the years. It's going on sixty-two years since it was first released and yet, it is **timeless**.

In the previous chapters, we required the respective algorithms to extract "The Wizard of Oz" in the first sentence and the last "it" in the second sentence as they are annotated as targets in the gold standard. However, if one wants to extract what the opinion in the second sentence is actually about, one has to resolve the anaphoric reference of "it" to "The Wizard of Oz" from the previous sentence.

In the opinion mining context, the extraction of such anaphoric opinion targets has been noted as an open issue multiple times in previous research (Zhuang et al., 2006; Hu and Liu, 2004a; Nasukawa and Yi, 2003). Some annotation studies have just observed the occurrences of anaphora as opinion targets, but did not annotate them (Somasundaran et al., 2008), while others do so (Kessler and Nicolov, 2009; Zhuang et al., 2006). The anaphoric reference of opinion targets is not a marginal phenomenon, since Kessler and Nicolov (2009) report that in the data they annotated, 14% of the opinion targets are pronouns. However, to the best of our knowledge the task of resolving anaphora to mine opinion targets has not been addressed and evaluated before in the literature.

It can therefore also be necessary to analyze more than the content of one individual sentence when extracting opinion targets. In this chapter, our goal is to take the additional step and require that our algorithmic approach also identifies the correct antecedent, given that an opinion target is an anaphor. We will investigate whether anaphora resolution can be successfully integrated into an opinion mining algorithm and whether we can achieve an improvement regarding the opinion target extraction in doing so. This chapter is structured as follows: In Section 5.1, we will discuss the related work on anaphora resolution in the context of opinion mining and

other NLP tasks and describe our system which extracts opinion targets while also resolving anaphoric targets to their antecedents. We will present an evaluation of the algorithm in Section 5.2, which we performed on the "movies", "web-services", "cars" and "cameras" datasets also employed in Chapters 3 and 4.

## 5.1 Algorithms

To the best of our knowledge, there is only one system which integrates coreference information in opinion mining. Anaphora resolution can be regarded as a subproblem of coreference resolution. While anaphora resolution focuses on the identification of the antecedents of pronomial references, coreference resolution aims at the identification of two or more noun phrases which refer to the same entity. The algorithm by Stoyanov and Cardie (2008) identifies coreferring opinion targets in newspaper articles. A candidate selection or extraction step for the opinion targets is not required, since they rely on manually annotated targets and focus solely on the coreference resolution. However, they do not resolve pronominal anaphora. In the following, we will outline how anaphora resolution algorithms have been successfully employed in other NLP tasks. We will then discuss the related work on anaphora resolution and how we integrated several algorithms in a baseline opinion mining system.

### 5.1.1 Anaphora Resolution to Enhance NLP Tasks

Vicedo and Ferrández (2000) successfully apply anaphora resolution to the tasks of question answering and information retrieval. In the information retrieval task, they substitute the resolved anaphora by their antecedents. Thereby, the term frequency of the antecedents is increased, which can positively influence the relevance ranking of documents which they occur in. In the question answering task, they do not explicitly substitute anaphora by their antecedents. Only during the calculation of a term's weight, the anaphora are virtually replaced by their antecedents. This is due to the fact that the relevant answers are extracted by calculating the cosine similarity to the respective question. They conclude that for both tasks a high-precision anaphora resolution is required in order to achieve an overall improvement in performance. In the information retrieval task, the anaphora resolution is most important when terms from the query are anaphorically referenced in the documents. In question answering, not only anaphora resolution, but also coreference resolution is of great importance. In some cases, coreference resolution can also allow for factual inference e.g. in

(5.2) ... Bill Clinton ... the president said ...

which is a useful feature for the question answering task. They perform their evaluation on a Spanish document collection, and the anaphora resolution algorithm was also specifically designed for Spanish.

Steinberger et al. (2005) improve the results of a summarization algorithm with coreference resolution. They employ the MARS algorithm (Mitkov, 1998) from the GuiTAR toolkit (Poesio and Kabadjov, 2004) for coreference resolution. They

observe that in this task the way in which the anaphoric information is used matters: By replacing anaphora with their respective antecedent, the authors do not reach any improvements in the summarization results. However, the authors suspect that this is due to the fact that they employ the LSA algorithm for the calculation of relevant terms for the summarization. By adding the terms from their anaphora chain to each of them, a different LSA clustering is achieved which in turn leads to better summarization results.

### 5.1.2 Algorithms for Anaphora Resolution

As pointed out by Charniak and Elsner (2009), there are hardly any freely available systems for anaphora resolution. Although Charniak and Elsner (2009) present a machine learning-based algorithm for anaphora resolution, they evaluate its performance in comparison to three non machine learning-based algorithms, since those are the only ones available. They observe that the best performing baseline algorithm (OpenNLP[1]) is hardly documented. The algorithm with the next-to-highest results in Charniak and Elsner (2009) is MARS (Mitkov, 1998) from the GuiTAR toolkit. This algorithm is based on statistical analysis of the antecedent candidates (a more detailed description follows below). Another promising algorithm for anaphora resolution employs a rule-based approach for antecedent identification. The CogNIAC algorithm (Baldwin, 1997) was designed for high-precision anaphora resolution (a more detailed description follows below). This approach seems appropriate for our opinion mining task, since in the dataset used in our experiments only a small fraction of the total number of pronouns are actual opinion targets (see Table 5.2). In the following, we will outline both algorithms for anaphora resolution. Both algorithms follow the common approach that noun phrases are antecedent candidates for the anaphora.

#### MARS

MARS (Mitkov, 1998; Mitkov et al., 2002) is a knowledge-poor anaphora resolution algorithm. It is based on a set of boosting and impeding indicators, which are employed for antecedent selection. The boosting indicators assign a positive score to a noun phrase and reflect a positive likelihood that this is the antecedent of the pronoun under analysis. The impeding scores reflect a low confidence that a given noun phrase is the antecedent of the pronoun which is to be resolved accordingly. The indicators were empirically defined and consider salience (definiteness, givenness, indicating verbs, lexical reiteration, section heading preference, non-prepositional noun phrases), structural matches (collocation, immediate reference) and referential distance. The candidate antecedents are selected within a distance of two sentences to the anaphora, and a check for gender and number agreement is performed. The algorithm identifies pleonastic occurrences of *"it"* with a set of more than 30 features which are a mixture of lexical and grammatical heuristics. An overall score is calculated for each antecedent candidate of a given pronoun, and the one with the highest score is selected.

[1] `http://opennlp.sourceforge.net/`

We employ the implementation of the MARS algorithm from the GuiTAR toolkit for anaphora resolution (Poesio and Kabadjov, 2004).

### CogNIAC

CogNIAC (Baldwin, 1997) is also a knowledge-poor approach to anaphora resolution. It is based on a set of six empirically defined rules, which are successively applied to the pronoun which is to be resolved. The rules are ordered according to their confidence and e.g. take in account whether a candidate antecedent is unique in the discourse or the current sentence. As soon as one rule matches an antecedent candidate of a pronoun, it is resolved. All rules contain the requirement of "antecedent uniqueness" as one feature. This means that the candidate antecedent is the single possible antecedent for a given pronoun. If this requirement is not met, the rule will not match. If none of the rules match or if two or more equivalent candidates are available, a pronoun is left unresolved. This strategy reflects the intention of the algorithm design of being sensitive to ambiguity and should lead to high precision. Since they are rather compact, we will quote the actual set of rules from Baldwin (1997) in the following for a better understanding of the algorithm:

1. **Unique in Discourse**: If there is a single possible antecedent $i$ in the read-in portion of the entire discourse, then pick $i$ as the antecedent.

2. **Reflexive**: Pick nearest possible antecedent in read-in portion of current sentence if the anaphor is a reflexive pronoun.

3. **Unique in Current + Prior**: If there is a single possible antecedent $i$ in the prior sentence and the read-in portion of the current sentence, then pick $i$ as the antecedent.

4. **Possessive Pro**: If the anaphor is a possessive pronoun and there is a single exact string match $i$ of the possessive in the prior sentence, then pick $i$ as the antecedent.

5. **Unique Current Sentence**: If there is a single possible antecedent $[i]$ in the read-in portion of the current sentence, then pick $i$ as the antecedent.

6. **Unique Subject/ Subject Pronoun**: If the subject of the prior sentence contains a single possible antecedent $i$, and the anaphor is the subject of the current sentence, then pick $i$ as the antecedent.

We employ the implementation of the CogNIAC algorithm as provided by the LingPipe toolkit[2]. We extended the available CogNIAC implementation to also resolve *"it"*, *"this"* and *"that"* as anaphora candidates since off-the-shelf it only resolves personal pronouns.

[2] `http://alias-i.com/lingpipe/`

**Baseline Opinion Mining Algorithm**

For our evaluation of the anaphora resolution for opinion target extraction, we naturally require an opinion mining algorithm into which we can integrate the anaphora resolution component. Theoretically all algorithms from Chapters 3 and 4 are good candidates, however the best results were reached by the supervised approaches from Chapter 4. Even though our CRF-based approach outperformed the Zhuang et al. (2006) baseline in our single-domain evaluation in Section 4.2.1, the algorithm design by Zhuang et al. (2006) has the following two advantages regarding a possible integration of an anaphora resolution component.

**Target Candidates:**
During the training phrase, the algorithm directly learns the target candidates. While this may lead to a loss of recall with unseen targets in the testing data, we can also directly learn the antecedents of the anaphora during training. In doing so, we can narrow down the set of antecedent candidates during the anaphora resolution process and thereby hopefully increase the precision. Consider the following example sentences, opinion targets are underlined and opinion expressions are shown in boldface:

(5.3) This film made it in the top ten (8 it reached).
So once televised on Sky Movies.
I watched the opening but I was rudely interrupted by my mate and I had to go out for an hour missing the parts but I am glad I did but I tell you about it later but the next time it was televised I did get to watch and it was **brilliant** though the story is a little bit **weak**.

The target of "brilliant" in the third sentence is the pronoun "it" which refers to "film" in the first sentence. If the noun phrase from the previous sentence "Sky Movies" is not in the target candidate list, we can filter it out from the possible antecedent candidates. Such a pre-filtering would require an additional extension of our CRF-based approach as the target candidates are learned implicitly by the algorithm and not directly added to a candidate list.

**Dependency - POS paths:**
The second feature of the Zhuang et al. (2006) baseline which enables a good integration with an anaphora resolution component is the approach of learning dependency - pos paths for the identification of opinion expression and opinion target pairs. Consider the following example sentences:

(5.4) I felt there was little that was original in this movie.
It was a weird mix of Stephen King, Wizard of Oz and Alice in Wonderland.

A common challenge for anaphora resolution algorithms are pleonastic occurrences of "it", as it is the case in the second sentence. The "it" at the beginning of the sentence does not have an antecedent since it is pleonastic. Given that there were no errors during the annotation, such an "it" should therefore never occur as an opinion target. Hence the algorithm by Zhuang et al. (2006) will not learn a

dependency - pos path which might falsely connect "it" and the possible opinion expression "weird". Even if the anaphora resolution falsely resolves the pleonastic "it" to e.g. "movie" of the previous sentence, the algorithm should not extract that false target candidate. Again, this feature would require an extension of our CRF-based approach, since its design is entirely different. Since the CRF-based approach is not a lexicalized algorithm, it does not directly learn the opinion target candidate strings from the training data. Instead, it performs the sequence labeling task (see Section4.1.2), in which the algorithm identifies the opinion targets based on the observed features and the learned model.

For these reasons, we decided to employ the algorithm by Zhuang et al. (2006) as a baseline in our experiments. An integration with our CRF-based approach is technically possible, but since our primary goal is to investigate the effects of an anaphora resolution component on the opinion target extraction, we believe that the algorithm by Zhuang et al. (2006) is a sufficiently sophisticated approach.

## 5.2 Experiments and Results

In the following, we elaborate on how we extended the opinion mining algorithm presented in Section 4.1.1 to integrate anaphora resolution. In the first step, we add the antecedents of the pronouns annotated as opinion targets to the opinion target candidate list. Furthermore, we extract the dependency paths connecting pronouns and opinion words and add them to the list of valid paths. When we run the algorithm, we extract anaphora which were resolved, if they occur with a valid dependency path to an opinion word. In such a case, the anaphor is substituted for its antecedent and thus extracted as part of an opinion target - opinion word pair.

There are three advantages of integrating anaphora resolution in opinion mining and especially in the algorithm by Zhuang et al. (2006). These are task-specific and do not necessarily hold in other tasks such as summarization or information retrieval:

- Not all pronouns need to be resolved (correctly), only the ones in sentences containing an opinion are required.
- The list of opinion target candidates prevents from using any antecedents which are completely out of context as possible targets.
- The learned dependency paths prevent from extracting pronouns which were never observed in a relation with an opinion word as opinion targets. This is of advantage with e.g. pleonastic *"it"* as in *"It was to be expected that Keanu Reeves' acting will be terrible."*

### 5.2.1 Datasets

In our experiments, we employ the same datasets as used in the previous experiments in Chapters 3 and 4 which we described in Chapter 2. In the following, we will present some statistics on the occurrences of pronouns in the respective datasets.

Anaphoric targets are annotated as such in each dataset, and for the total number of pronouns we counted their occurrences as identified by the Stanford Parser[3].

Table 5.1: Anaphoric Target Statistics

| | **movies** | **web-services** | **cars** | **cameras** |
|---|---|---|---|---|
| Targets | 7045 | 1875 | 8451 | 4369 |
| Anaphoric Targets | 712 | 297 | 505 | 335 |
| Pronouns | >38000 | >11050 | >9900 | >6000 |

As Table 5.1 shows, between $\sim 6\%$ and $\sim 15\%$ of the opinion targets are referred to by pronouns across the four datasets. Table 5.2 outlines detailed statistics about which pronouns occur as opinion targets. Overall only a small fraction of the pronouns which occur in the corpus refer to actual targets. We also observe that in the "movies" dataset the percentage of personal pronouns referring to humans (he, him, his, she, her, hers) is much higher than in the other three datasets. This is likely to be attributed to the domain, in which (human) characters, or actors playing them, are discussed, which is far less frequently the case when talking about web-services, cars or cameras. In the "other" category, we subsumed all annotated targets which were marked as being anaphoric, but none of the pronouns above. These could e.g. be misspellings, which are quite common in user-generated discourse.

Table 5.2: Pronouns as Opinion Targets

| | **movies** | **web-services** | **cars** | **cameras** |
|---|---|---|---|---|
| it | 370 | 101 | 348 | 199 |
| this | 87 | 67 | 20 | 40 |
| that | 0 | 10 | 10 | 9 |
| they | 22 | 63 | 42 | 34 |
| he | 59 | 2 | 9 | 2 |
| him | 6 | 0 | 1 | 0 |
| his | 26 | 0 | 0 | 2 |
| she | 13 | 1 | 8 | 0 |
| her | 11 | 0 | 3 | 1 |
| hers | 0 | 0 | 0 | 0 |
| other | 118 | 53 | 64 | 48 |

Note that the demonstrative pronoun *"that"* is rarely used an opinion target. The Givenness Hierarchy (Gundel et al., 1993) defines six cognitive statuses of referring expressions. These statuses rank the referent's assumed location in memory and attention state of the reader. If the referred entity is in the highest status ("in focus"), it is typically referred to by *"it"*. In the second highest status ("activated")

[3] `http://nlp.stanford.edu/software/lex-parser.shtml`

the pronoun *"this"* is used if it is the subject of the sentence. Only in the third status ("familiar") the pronoun *"that"* is used. The familiar status defines that the reader can associate a unique representation with the entity that is already in memory somewhere, perhaps long-term memory (Gundel et al., 1993). Apparently, in the context of reviews the authors mostly make anaphoric references in the higher statuses, which require that the referent has either been introduced in the text or is in the direct context.

### 5.2.2 System Configuration

To reproduce the system by Zhuang et al. (2006), we had to substitute the cast and crew list employed by them (see Section 4.1.1), since we could not reconstruct it. Instead, we utilized a standard named entity recognition component, successfully applied in previous research for the detection of peoples' names (Finkel et al., 2005). Again, we merge adjacent target candidates in a sentence to a multiword target and require targets to be extracted with the boundaries exactly matching.

The GuiTAR implementation of the MARS algorithm resolves anaphora to complete noun phrases including possible determiners. Since in the gold standard determiners are not annotated as parts of opinion targets, we remove them from the antecedents GuiTAR extracts before the evaluation.

### 5.2.3 Results

Table 5.3: Results of Baseline Including Anaphoric Targets

| **Dataset** | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|
| movies | 0.613 | 0.499 | 0.550 |
| web-services | 0.462 | 0.292 | 0.358 |
| cars | 0.240 | 0.395 | 0.298 |
| cameras | 0.388 | 0.449 | 0.416 |

Table 5.3 shows the results of the baseline algorithm. These results are comparable with the single-domain results of Section 4.2.1 (see Table 4.2). As evident from Table 5.3, both precision and recall are lower in this setting compared to the results we reached in Section 4.2.1. This was to be expected, as the baseline is not capable of resolving any anaphoric targets to their antecedents.

Table 5.4 shows the results of the target extraction using the MARS algorithm (Mitkov et al., 2002) for anaphora resolution. As evident from the Table, there is a general and consistent trend regarding the target extraction performance when MARS is employed for the anaphora resolution. The algorithm correctly extracts some of the anaphoric targets which results in an increase of recall compared to the baseline configuration from Table 5.3. However, the inclusion of the anaphora resolution component results in a loss of precision on all four datasets. This is especially problematic regarding the overall results on the "cars" and "cameras" datasets,

Table 5.4: Results of Baseline + MARS Including Anaphoric Targets

| Dataset | Precision | Recall | F-Measure |
|---|---|---|---|
| movies | 0.562 | 0.518 | 0.539 |
| web-services | 0.412 | 0.309 | 0.353 |
| cars | 0.209 | 0.410 | 0.276 |
| cameras | 0.341 | 0.474 | 0.396 |

since the precision is already considerably lower than the recall even in the baseline configuration. In total, the loss of precision outweighs the increase of recall which leads to a lower F-Measure on all datasets.

A manual inspection of the results has shown that the MARS algorithm indeed attempts to resolve every pronoun it encounters. Mitkov et al. (2002) state that their algorithm has a component to detect pleonastic occurrences of "it". However, in several processed samples we manually inspected we observed many cases in which MARS attempted to resolve such non-referential instances of "it". Maybe this phenomenon has to be attributed to the GuiTAR implementation of the algorithm (Poesio and Kabadjov, 2004) which was available to us.[4]

While we cannot assess the overall accuracy of the anaphora resolution on our datasets, the increase of recall indicated that at least for the anaphora which are actual opinion targets MARS yields decent results. However, the loss of precision shows that the strategy of resolving every pronoun can be quite damaging to the opinion mining setting as it can lead to many false positives. This is due to the fact that for our opinion mining algorithm every resolved pronoun is a new target candidate. Judging from our manual inspection of the output, we conclude that the extraction strategy of the baseline opinion mining algorithm even mitigates the number of false positives. The dependency - pos paths should in fact limit the amount of false positives, but as the algorithm learns several hundreds of these patterns for a given dataset (see Section 4.2.1), accidental matches are definitely possible.

Table 5.5: Results of Baseline + CogNIAC Including Anaphoric Targets

| Dataset | Precision | Recall | F-Measure |
|---|---|---|---|
| movies | 0.616 | 0.506 | 0.556 |
| web-services | 0.477 | 0.301 | 0.369* |
| cars | 0.241 | 0.400 | 0.301 |
| cameras | 0.392 | 0.469 | 0.427* |

[4]The source code of the GuiTAR library was not available to us, therefore we could not investigate possible causes for these cases.

As shown in Table 5.5, the off-the-shelf CogNIAC algorithm extended to resolve also "it", "this" and "that" already yields significant improvements over the baseline regarding F-Measure[5] on the "web-services" and the "cameras" datasets. By including CogNIAC for the anaphora resolution, we manage to increase both the precision and recall of the target extraction on all datasets. The recall of the target extraction is not as high as for the configuration in which we employ MARS for the anaphora resolution. However, since CogNIAC does not decrease the precision of the target extraction, the overall results regarding F-Measure are better compared to our configuration with the MARS algorithm.

Although the results of the opinion target extraction using CogNIAC algorithm are already promising, we identified a few sources of errors in a preliminary error analysis. We propose three extensions to the algorithm which are on the one hand possible in the opinion mining setting and which are on the other hand possible due to features of the discourse type we are working with:

**Extensions of CogNIAC**

1. Our first extension is based on the observation that the Stanford Named Entity Recognition (Finkel et al., 2005) algorithm is superior to the PERSON / LOCATION / ORGANIZATION detection of the (MUC6 trained) CogNIAC implementation. We therefore filter out PERSON antecedent candidates which the Stanford Named Entity recognizer detects for pronouns "it", "this" and "that', and LOCATION & ORGANIZATION candidates for the personal pronouns referring to animate males or females ("he", "she", "they", ...). In doing so, we optimize the input for the anaphora resolution process.

2. The second extension exploits the fact that product or movie reviews exhibit certain contextual properties. They are gathered and to be presented in the context of one particular entity (e.g. movie, web-service, car, camera, ...). The context or topic under which it occurs is therefore clear to the reader and it is therefore not required to explicitly introduce them in the discourse. A review is typically presented under the description of the entity it refers to. This is equivalent to the situational context we often refer to in a dialogue. In e.g. the movie reviews the authors often refer to the movie or film as a whole by the pronoun "it". We exploit this by adding an additional rule which resolves a candidate impersonal or demonstrative pronoun to the overall dataset topic (movie, web-service, car, camera) if there is no other (matching) antecedent candidate in the previous two sentences.

3. The rules by which CogNIAC resolves anaphora were designed so that anaphora which have ambiguous antecedents are left unresolved. This strategy should lead to a high-precision anaphora resolution, but at the same time it can have a negative impact on the recall. In the opinion mining context, it happens quite frequently that the authors comment on the entity they want to criticize in a series of arguments. This is a typical case in which people make use of anaphora, and we try to solve cases of antecedent ambiguity by analyzing

[5]Significance of improvements was tested using a paired two-tailed t-test and $p \leq 0.05$ (*), $p \leq 0.01$ (**), and $p \leq 0.005$ (***).

the opinions: If there are ambiguous antecedent candidates for a pronoun, we check whether there is an opinion uttered in the previous sentence. If this is the case and if the opinion target matches the pronoun regarding being animate or inanimate, matches in gender and number, we resolve the pronoun to the antecedent which was the previous opinion target.

In the following, we will evaluate our three CogNIAC extensions, again in the task of opinion target extraction in the same experimental setting as described above.

Table 5.6: Results of Baseline + Extended CogNIAC Including Anaphoric Targets

| **Dataset** | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|
| movies | 0.643 | 0.528 | 0.580* |
| web-services | 0.506 | 0.320 | 0.392* |
| cars | 0.246 | 0.408 | 0.307 |
| cameras | 0.400 | 0.478 | 0.436* |

As evident from Table 5.6, our three extensions yield significant improvements[6] regarding both precision and recall compared to the baseline system with the off-the-shelf CogNIAC algorithm on three of the four datasets. In a set of additional experiments we analyzed the influence of the individual extensions and their combinations.[7] We observed that our extensions are complementary and that the best results regarding F-Measure are reached if they are combined. The improvements we yield on the "cars" dataset are not statistically significant. In the configurations with MARS for the anaphora resolution, this dataset also exhibited the biggest decrease of F-Measure. As shown in Table 5.2, in the "cars" dataset the pronoun "it" is by far the most frequent one. Since this pronoun can occur pleonastically, it is more difficult to resolve and none of our extensions actually properly address this problem. We assume that therefore our extensions have less positive impact on this dataset.

### 5.2.4 Error Analysis

In order to evaluate our results in the broader context, we will calculate the upper bounds of precision and recall which an opinion target extraction system with a perfect anaphora resolution component can yield. As shown in Table 5.2, there are quite a few anaphoric opinion targets in each dataset which we could not attribute to any of the pronouns shown in that Table. In a manual inspection of both the annotated data and our system's output we have identified the following reasons:

- Errors in our preprocessing components: We have seen some cases in which our sentence splitting or tokenization components introduced errors which made

[6]Significance of improvements was tested using a paired two-tailed t-test and $p \leq 0.05$ (*), $p \leq 0.01$ (**), and $p \leq 0.005$ (***).

[7]Results not shown here due to the high number of feature combinations, please refer to (Jakob and Gurevych, 2010b).

it impossible for the part-of-speech tagger to detect a pronoun in a sentence. However judging from the sample output we have inspected, these errors seem to be rare.

- Spelling errors in the source documents: A very prominent example is the misspelling of "it's" as "its" in the source documents. These errors are to be expected when working with user-generated discourse, but this particular misspelling is even difficult to recognize for automatic spelling correction systems.

- Errors / artifacts in the annotation: There are some cases in which the annotation boundaries are just wrong, e.g. we have seen annotation spans which cover pronouns with whitespace and some arbitrary characters of the following word. These errors can just be corrected by manually verifying and perhaps consolidating the annotation. Furthermore, we have seen several cases in which coreferent targets were labeled as anaphoric, e.g. in one sentence "BMW Z5" is the opinion target and in a following sentence "car", which refers to the same entity via coreference. Again, a manual inspection of the annotation would be required to correct these artifacts.

For our calculation of the upper bounds, we manually filtered out all of these erroneous cases, since the anaphora resolution algorithms do not even receive them as input. The values are shown in Table 5.7.

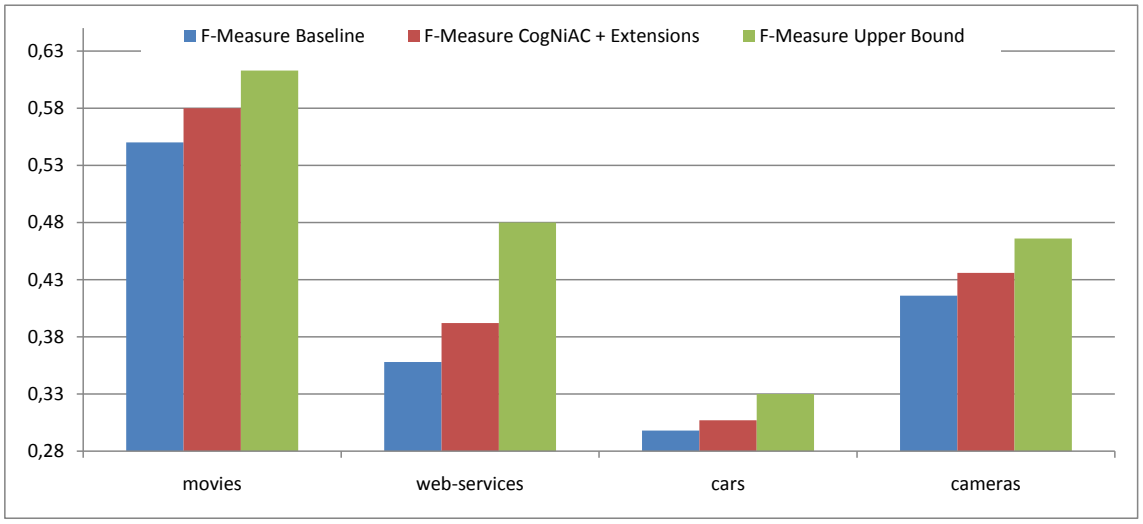
Table 5.7: Upper Bound for Baseline + Perfect Anaphora Resolution

| **Dataset** | **Precision** | **Recall** | **F-Measure** |
|---|---|---|---|
| movies | 0.645 | 0.584 | 0.613 |
| web-services | 0.555 | 0.423 | 0.480 |
| cars | 0.261 | 0.446 | 0.330 |
| cameras | 0.418 | 0.526 | 0.466 |

Figure 5.1 visualizes the results of the target extraction of the baseline system compared to our approach with the extended variant of CogNIAC and finally the upper bound. By comparing the maximally possible increase regarding F-Measure in Table 5.7 with the best results we reached with CogNIAC and our three extensions (see Table 5.6), we can determine how much of the theoretically possible increase we achieved. For the "movies" dataset, the upper bound regarding an increase of F-Measure is 0.063. Our extended version of CogNIAC yields an increase by 0.030 regarding F-Measure which corresponds to 47.6%. On the "web-services" dataset, the maximal possible increase of F-Measure is 0.122, while our approach reached an increase by 0.034 which corresponds to 27.8% of the theoretically possible improvement. On the "cars" dataset, the increase of F-Measure with perfect anaphora resolution is 0.032, while our extended CogNIAC system yields an improvement by 0.009 which is 28.1% of the maximal possible increase. Finally, on the "cameras" dataset the maximal possible increase regarding F-Measure is 0.05, and our approach

yields an increase of 0.02 which corresponds to 40%. Our results are hence quite promising on the "movies" and "cameras" datasets, considering that in the evaluation by Charniak and Elsner (2009), the unsupervised algorithms have an anaphora resolution accuracy of 0.534 (CogNIAC) and 0.529 (MARS) on newspaper articles.

Figure 5.1: Target Extraction Baseline vs. Baseline + Extended CogNIAC vs. Upper Bound




In the following, we will present our analysis of the different classes of errors regarding the anaphora resolution. We have identified three types of errors in the anaphora resolution which can subsequently result in errors in the target extraction:

**1. An anaphoric target is resolved to an incorrect antecedent**

We will visualize a scenario for this error class in the following example. The anaphoric target is underlined, the corresponding correct antecedent is shown in boldface, the incorrect antecedent is shown in square brackets and the opinion expression is highlighted by italics:

(5.5) I saw **Lord of the Rings** for the first time in the last [month]. <u>It</u> completely *blew my mind*.

The scenario for this error class is that there is a sentence which contains an anaphoric target ("It") and the anaphora resolution algorithm attempts to resolve it, but selects an incorrect antecedent ("month"). If we assume that the opinion mining algorithm has learned a dependency - pos path which connects the opinion expression ("blew my mind") and the anaphoric target ("It"), this scenario will even lead to two errors in the extraction: The opinion mining algorithm will select the incorrect antecedent as the opinion target for this sentence ("month") and the actual target ("Lord of the Rings") will not be extracted. This results in both a false positive and a false negative during the target extraction. The incorrect antecedent is the false positive and since only one antecedent is selected for a given pronoun,

the actual correct antecedent will not be extracted and therefore is a false negative. Such an error will therefore result in a loss of both precision and recall.

A solution to these kinds of errors can clearly only be found in a better anaphora resolution. An anaphora resolution algorithm with higher accuracy would be required. Charniak and Elsner (2009) present a supervised approach for anaphora resolution which reaches an accuracy of 68.6% on newspaper articles. How well this algorithm performs on our user-generated discourse could be investigated in future work.

**2. An anaphoric target is not resolved at all**

There are three reasons why an anaphoric opinion target might not be resolved to any antecedent:

1. A spelling error in the source document or an error in our NLP preprocessing, as mentioned in the beginning of this Section.

2. CogNIAC is employed for the anaphora resolution and due to antecedent ambiguity, the algorithm does not attempt to resolve the pronoun.

3. The baseline opinion mining algorithm has not learned a matching dependency - pos path connecting the anaphoric target and the opinion expression in that sentence.

If one of the above problems occur, a given anaphoric target will not be extracted, resulting in a false negative. This will lead to a loss of recall during the target extraction. However, this type of error is actually favorable over an error of class 1 described above as it only negatively affects the recall. Since the MARS algorithm always attempts to resolve every pronoun it encounters, errors of class 1 are much more likely compared to when CogNIAC is employed for the anaphora resolution.

A possible solution to these types of errors could be the integration of a spelling correction component in the preprocessing. In doing so we might be able to correct misspellings of anaphora and in turn increase the number of anaphoric target candidates. The CogNIAC algorithm will not resolve an anaphoric target if there are ambiguous antecedent candidates. If we could narrow down the number of antecedent candidates, we might be able to increase the number of anaphoric targets which CogNIAC attempts to resolve. This might be achieved by also learning a set of antecedent candidates during the target learning step of the baseline algorithm.

**3. A pronoun which is actually not an opinion target is falsely extracted as such**

We will visualize a scenario for the following error class in the following example. Again the opinion expression is shown in italics and the opinion target is shown in boldface. The correct pronoun - antecedent pair is shown in square brackets, which is resolved as such by the anaphora resolution algorithm.

(5.6) I got the [G5] as a christmas present from my parents. And [it]'s got lots of those *great* **Canon camera features**.

Note that in this example there is actually no anaphoric target, as "great" in the second sentence refers to "Canon camera features". However, by chance there might be a dependency - pos path connecting "great" and "it" which matches a pattern that the opinion mining algorithm has learned from another sentence. In this case, the algorithm will (also) extract G5 as an anaphoric target via the "it" from the second sentence. This type of error can hence even occur if there are no errors at all made by the anaphora resolution algorithm.

Such errors are obviously introduced by the design of the baseline opinion mining algorithm. Switching to a different baseline system could solve these types of errors. However as we have observed in the analysis of the results of the MARS algorithm, the dependency - pos paths are also quite valuable since they can prevent the erroneous extractions of pleonastic "it" as opinion targets.

## 5.3 Chapter Summary

In this Chapter, we have shown that by extending an opinion mining algorithm with anaphora resolution, significant improvements regarding the opinion target extractions can be achieved. We have shown that the CogNIAC algorithm, which was designed for a high-precision anaphora resolution outperforms the MARS algorithm in the task of extracting anaphoric targets. We presented a set of extensions to the CogNIAC algorithm, which address some of the initial challenges we observed and our extensions significantly improved the opinion target extraction. The MARS algorithm does not yield any improvements regarding F-Measure in the opinion target extraction, since it generates too many incorrect opinion target candidates. The algorithm creates many false positives, which are not filtered out by the dependency paths employed in the algorithm by Zhuang et al. (2006) for finding valid opinion target - opinion word pairs.

An anaphora resolution component could also be employed in other opinion mining algorithms which aim at identifying opinion targets with a statistical analysis such as the Likelihood Ratio Test presented in Chapter 3. Vicedo and Ferrández (2000) have successfully modified the statistical relevance ranking of terms in their documents by replacing anaphora with their antecedents. This approach can also be employed for opinion mining algorithms, which select the opinion target candidates by means of a relevance ranking (Hu and Liu, 2004a; Yi et al., 2003).

In future work, we might investigate how machine learning-based algorithms for anaphora resolution perform in the opinion target extraction task. Yang et al. (2006); Haghighi and Klein (2007); Charniak and Elsner (2009) all presented supervised approaches which yield better results in the anaphora resolution on newswire (F-Measure of up to 0.706 (Haghighi and Klein, 2007)). In the gold standard employed in our experiments, only anaphora which are opinion targets are annotated with their respective antecedent. It should be investigated whether this number of instances is enough to train a machine learning algorithm and how to process anaphora which are not resolved during the training phase. Alternatively, one could employ the model trained on documents from a different domain, e.g. newswire, and analyze how it performs on user-generated discourse. This is again related to the cross-domain extraction task we studied in Chapter 4.

# Chapter 6

# Opinion Mining to Improve Recommendation Systems

One of the key characteristics of Web 2.0 is that it allows internet users to share with other users their viewpoints and opinions about almost everything. Hearing another person's substantiated opinion can be of practical benefit when it comes to deciding whether or not to invest time, money or effort into something. This is one of the driving forces behind the increasing success of community web sites which allow registered users to write and read reviews about commercial products such as books, music, movies, or consumer electronics devices such as e.g. digital cameras or cell phones.

User ratings often consist of a free-text review and an overall rating. The present work focuses on the domain of movies. In this domain, the overall rating often comes in the form of a *star* rating. Collected user ratings can be organized by movie and presented to users who are interested in other users' opinions about a particular movie. Increasingly often, the data is also used for the creation of personalized recommendations, in which users are proactively presented with movies which they probably like. Most recommendation systems only take into account the obligatory star ratings and some simple descriptive movie features (e.g. genre) (Takacs et al., 2007; Yu et al., 2005) and leave completely unused the wealth of information that is included in the free-text reviews.

In opinion mining, a lot of work has already been done on extracting fine-grained opinion expressions from free text (Gamon et al., 2005; Popescu and Etzioni, 2005; Zhuang et al., 2006). It is consequential, therefore, to bridge the gap between opinion mining and recommendation systems and to go beyond the information conveyed by the star ratings by also exploiting free-text user reviews for recommendation. We propose to do this by employing phrase-level opinion mining on free-text movie reviews for the identification of positively and negatively opinionated user statements, and by incorporating this information into the state-of-the-art recommendation system HYRES (Lippert et al., 2008). Opinionated user statements consist of the opinion-bearing expression (e.g. an adjective like "poor" or "beautiful") and the opinion *target*, i.e. what is being commented on.

The fundamental rationale of our approach is that two important types of information can be extracted from the free-text reviews:

1. The correlation of the overall star rating with the individual aspect-related

opinions shows the influence on the star rating that a given movie aspect has for a user.

2. The overall number of opinions regarding a certain movie aspect cluster reveals how important that aspect is to a user.

We argue that it is desirable, e.g., to also recommend movies with only a mediocre star rating to a user, if they are rated well regarding one or more aspects which are of high interest to that user. Vice versa, a well-rated movie should not be penalized for poor ratings regarding aspects which are known to be of low importance for a given user. Our goal is to achieve this by being able to model the user's preferences with a very fine granularity. Assume for example that a user $U$ despises a certain actor $A$ and that this actor's performance repeatedly ruined an otherwise perfectly enjoyable movie for $U$. Now it is likely that $U$ gives such a movie a bad or mediocre rating and expresses his disappointment in a review. In this review, $U$ utters his disdain for $A$'s acting, but also positively mentions aspects he liked about the movie, e.g. the cinematography by director $D$ or the soundtrack etc. If we could design a system which is capable of extracting the individual opinion uttered in the review and their respective targets, we could use them as additional "ratings" which the user gives to these aspects of the movie.

The common approach when modeling a recommendation system nowadays is to include as much additional information about the rated entity (e.g. "$A$ stars in movie $M$", "$D$ directs movie $M$", "$S$ created soundtrack for movie $M$") and also about the user who gives the rating. During the training phase, the recommendation system then propagates the overall star rating, which the user has given to the movie, onto the different aspects which have been included during the modeling. However, this approach disregards that the user can have quite faceted opinions about the individual aspects, some better some worse, which ultimately form the overall impression. Consequently, the recommendation system will need several rating instances to be able to detect patterns such as "if $A$ stars in a movie, this leads to a mediocre rating, therefore do not recommend movies with $A$ to $U$". With our approach, we strive to overcome this limitation, since with the ratings which we extract from the opinions the recommendation system will not have to infer such information by correlation, but instead directly receives input regarding the individual aspects.

There is however a complexity problem which we introduce with our approach. Since there are no constraints regarding the aspects of a movie which the users can comment on in a free-text review, each aspect on which an opinion is uttered would introduce a new dimension in the representation of a movie. Therefore, we have to somehow condense this information before it can be incorporated into the recommendation system. We do this by mapping each automatically extracted opinion target to one or more pre-defined descriptive categories corresponding to movie-related concepts such as "acting" or "soundtrack". We use the term *movie aspect cluster* to refer to the result of these mappings. We accumulate all opinionated statements for each movie aspect cluster and provide them to the recommendation system together with the original star rating. The recommendation system is then used as a black box for extrinsically evaluating the effects of the automatically extracted information.

## 6.1 Recommendation Systems

Recommendation systems are algorithms which attempt to predict items (e.g. movies, music, books) in which a user may be interested, given some information about the item and/or the user's profile. Content-based algorithms only use information about the items. Items are recommended that are most similar to the items the current user likes. However, the item description cannot capture all relevant aspects of the item and the user's perception of it (e.g. mood). Furthermore, following content-based recommendations the user will stick to his usual preferences. Only items will be recommended that are similar to those already rated. Collaborative filtering overcomes this limitation by making use of the user's personal preferences and information, e.g. previously bought items, ratings or contacts. Collaborative filtering algorithms make use of the collaborative effect and recommend items that have been highly rated by likeminded users.

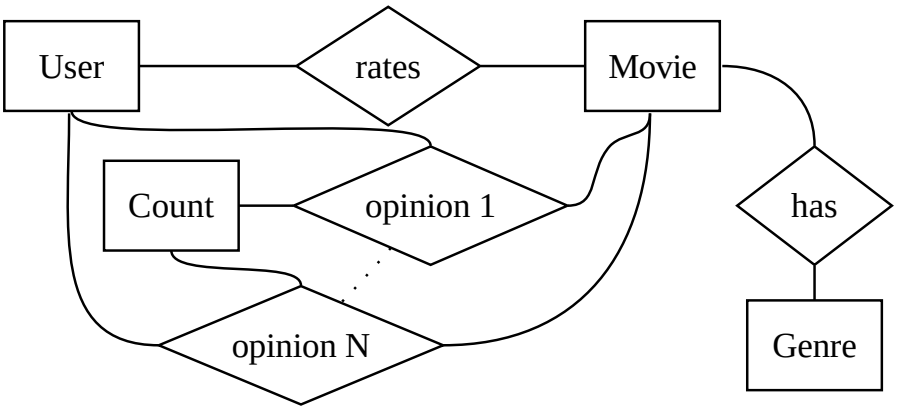
These two complementary recommendation approaches are combined in the hybrid and platform independent framework HYRES (HYbrid REcommendation System). HYRES implements the MRMF algorithm (Lippert et al., 2008), which can handle an arbitrary number of entity types and relation types in a given domain and exploits multiple relation types simultaneously. Apart from its high accuracy, the system also exhibits a high performance even on huge data sets (e.g. the NetFlix data set). We chose HYRES as the basis for our experiments because it can easily handle more dimensions and any number of entities and relations. Furthermore, the extension of HYRES is straightforward for us since it is a Java library and our development is also done Java-based.

A natural way of representing relational data is an entity relationship model. Our example data set can be described by an ER diagram as depicted in Figure 6.1. Involved entities are *User*, *Movie*, *Genre* and *Count*, where *Count* denotes the discretized average number of given opinions of a certain type for a single movie. For example, "I like actor $A$" and "I dislike actor $B$" are both opinions of opinion type *acting*. Relations that have to be considered in the collaborative filtering model are "User rates Movie", "Movie has Genre", and for each opinion type $N$ the n-ary relation, "User has opinion $n$ about Movie" averaged over "Count opinions" of that type.

Several aspects had to be taken into account to transfer the ER diagram to a multi-relational collaborative filtering model. The model shown in Figure 6.2 illustrates the full collaborative filtering model. The n-ary opinion type relations between *User*, *Movie* and *Count* had to be decomposed by reification since MRMF only handles binary relations. Each opinion type relation is modeled as an entity by itself. This corresponds semantically to split the *User*s into their different roles: the rating-role and a role for each opinion type. However, by modeling the *User*s as separate entities, the knowledge about the same identity of individual users in different roles is lost. This is compensated by introducing a new sparse relation, *sim*, between the user roles, mapping individual users on their different roles. The model contains the following five relation types modeled as bipartite adjacency matrices.

1. The sparse matrix $\mathbf{rates} \in \mathbb{R}^{u \times m}$ contains the overall star rating of users for movies (1 to 10), where $u$ is the number of users and $m$ is the number of

Figure 6.1: Entity Relationship Diagram of Dataset




movies.

2. $N$ sparse matrices $\mathbf{hasOp}_N \in \mathbb{R}^{u \times m}$ contain the averaged values for opinion type $N$ of users for movies (1 to 10).

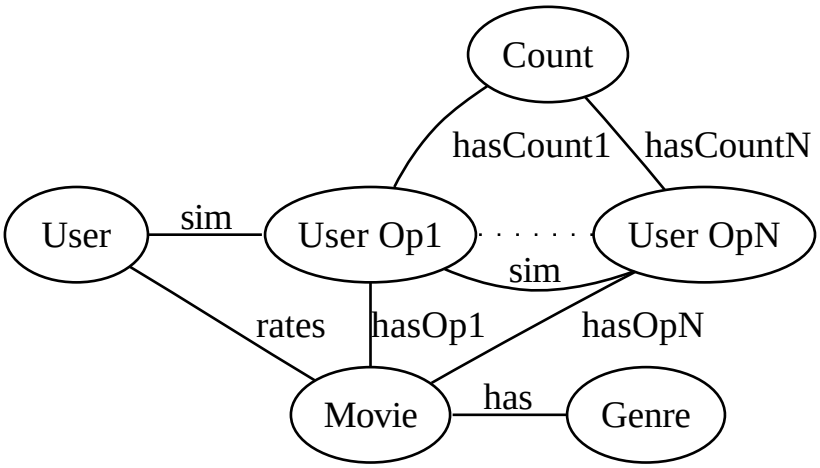
3. The dense binary matrix $\mathbf{has} \in \{0,1\}^{m \times g}$ maps movies on genres, where $g$ denotes the number of genres. All known genre relations are labeled with 1 whereas unknown genre affiliation is modeled as 0.

4. $N$ dense binary matrices $\mathbf{hasCount}_N \in \{0,1\}^{u \times c}$ map the averaged opinion on the discretized number of given opinions $c$.

5. $N$ sparse matrices $\mathbf{sim}_N \in \mathbb{R}^{u \times u}$ map the similarity between users in their different roles. Note that only the matrix diagonal is filled with the known similarity of 1 and unknown similarities are modeled as 0.

The above entities and relations are supplied to the system as feature vectors. Extending the collaborative filtering model, i.e. adding new features to the model, only requires manually editing the model file, which is a one-time effort, and appending the values for the new features to the existing feature vectors. In our experiments we also investigate sub-models containing only a subset of the full model. The smallest sub-model, "User rates Movie", consists of two entities and one relation, whereas the full-blown collaborative filtering model for 20 opinion types results in 24 entities and 62 relations. In order to make all models comparable, we abstained from optimizing free parameters for each model, but fixed the free parameters to reasonable values acceptable for all models. Free parameters include learning rate, regularizer rate and the maximal number of learning epochs. For more information on the default parameters and settings see Lippert et al. (2008).

Figure 6.2: HYRES Collaborative Filtering Model




## 6.2 Clustering Approaches in Opinion Mining

A second aspect which we cover in this work aims at clustering the identified opinions by topics. In previous research such a clustering was employed in order to separate opinion bearing words from the respective targets (Mei et al., 2007) or in order to group opinions regarding the same target and sentiment orientation together (Lu and Zhai, 2008). Such a topic clustering can also be employed in order to separate documents from different domains and cluster the opinions regarding possible subtopics therein (Titov and McDonald, 2008a). We perform the movie aspect clustering in order to create usable input for the recommendation system, but it can also be used to create a more useful output of an opinion mining system for the end-user by creating summaries of the reviews (Blair-Goldensohn et al., 2008).

Various technological approaches of recommendation systems have been described and compared in detail, e.g. in Breese et al. (1998); Schafer et al. (2007); Herlocker et al. (2004). All the described predictive models focus on a single relation type (*rates*) between two entity types (*user*, *item*). Matrix factorizations such as Singular Value Decomposition (SVD) have recently been applied to relation prediction. The maximum margin matrix factorization (MMMF) introduced in Srebro et al. (2005) is a matrix factorization approach based only on the known matrix entries. Unfortunately, the MMMF model is hardly scalable. A way to make the model more scalable is to minimize the objective by using gradient descent methods. In Takacs et al. (2007), one of the favored approaches in the Netflix Prize[1], a simple gradient descent method was applied. Recently some unsupervised approaches (Long et al., 2006b,a) have been proposed to deal with graph clustering problems on multi-relational domains. Lippert and Weber Lippert et al. (2008) introduced a multi-relational matrix factorization (MRMF) which is an extension of low-norm

[1] `http://www.netflixprize.com/`

matrix factorization to multi-relational domains where the involved relation types are usually highly correlated. To the best of our knowledge, there is only one approach of integrating opinion mining with a recommendation system described in the literature (Aciar et al., 2007). However, the case study presented requires users to formulate their demands in the form of a query, which is then matched to opinions uttered towards the respective aspects in other users' reviews. The present work, in contrast, strives to extract user preferences automatically from ratings and existing free-text reviews.

## 6.3 Extracting Opinions to Improve Movie Recommendations

In the following sections, we will describe the three different approaches to movie aspect identification and clustering that we have experimented with, and in Section 6.3.2 we will elaborate on our opinion extraction pipeline.

### 6.3.1 Movie Aspect Identification & Clustering

As already outlined in the beginning of this Chapter, the set of movie aspects that a user can comment on in his or her review is in principle unconstrained. In order to integrate the opinions expressed in a given review into the recommendation system, they need to be represented in a more compact way. We do this by mapping each identified opinion target to one or more pre-defined movie aspect clusters, and by computing several overall numerical values for each cluster. The composition of the movie aspect clusters thus has a major impact on the recommendation system, which is why we tried several ways of creating them.

#### Manual Clustering

In a first attempt, we created five medium-sized movie aspect clusters manually. In order to achieve this, we read the Wikipedia article on "*Film*" and identified the following key concepts regarding this topic: "*acting*", "*storyline*", "*cinematography*", "*soundtrack*", and "*production*". By analyzing the corresponding articles, we then identified for each category between five and 20 pertinent terms which we considered to be potential opinion targets. Excerpts of the resulting movie aspect clusters can be found in Table 6.1. We intentionally left out general terms such as "*movie*" or "*film*", since opinions regarding these terms do not refer to a certain movie aspect, but express the user's opinion on the movie as a whole. This information, however, is already given by the star rating. We treated opinions regarding individual actors and directors as related to the concepts "*acting*" and "*cinematography*", respectively. For this, we extracted 11015 actor names from the Wikipedia categories "American actors" and "American film actors", and 1171 director names from the category "American film directors".

Table 6.1: Manual Cluster Excerpts (Size in Brackets)

| **acting** (8) | **storyline** (15) | **production** (14) |
|---|---|---|
| actor | story | set |
| actress | beginning | scenery |
| acting | ending | costume |
| role | script | producer |
| cast | plot | crew |
| ⋮ | ⋮ | ⋮ |
| **soundtrack** (4) | **cinematography**(20) | |
| music | camera angle | |
| score | shot | |
| song | slow-motion | |
| soundtrack | director | |
| | editing | |
| | ⋮ | |

### Semi-automatic Clustering

The manual identification and clustering approach described above has two major disadvantages: Manually selecting terms for each of the five movie-related concepts by inspecting a resource such as Wikipedia is a very time-consuming task, and the recall can still be poor because the resource might not cover all of the terms that are used in the review corpus. We therefore also tried a semi-automatic clustering approach which used as input only the five manually defined key concepts. The semi-automatic clustering first identifies the potential movie aspects among all opinion target candidate terms. Some of these are then mapped to exactly one of the five categories, while others remain unmapped. Opinion target candidate terms are all terms in the review corpus (Table 6.4) after opinion expression (see Section 6.3.2) and stop word removal. We based the clustering on the notion of *semantic relatedness* between a candidate term and a cluster's key concept. Several approaches for measuring the semantic relatedness of terms have been suggested in the past of which Explicit Semantic Analysis (ESA) on Wikipedia represents the state-of-the art in several tasks (Gabrilovich and Markovitch, 2007). ESA is a vector based measure for the calculation of the semantic relatedness of two words. It requires a corpus $D$, which is employed to create the representation of the word meanings. The meaning of a given word $w$ is represented as a concept vector $\vec{d}(w) = (d_1, \ldots, d_N)$. Each element $d_i$ represents a document from $D$, whereas the value of $d_i$ corresponds to the number of occurrences of the word $w$ in this document. Gabrilovich and Markovitch (2007) employ Wikipedia as a corpus and its articles as documents. The underlying intention is, that each Wikipedia article describes a certain topic / concept. The size of the concept vector corresponds to the number of articles in Wikipedia $N$.

Each element of the concept vector $\vec{d}$ therefore corresponds to a Wikipedia article. The abovementioned "value" of $d_i$ is the tf-idf score (Salton and McGill, 1983, page 129) of the word $w$ in the current article $a_i$. If the word $w$ does not occur in a given Wikipedia article $a_i$, $d_i$ is set to 0. The entire vector $\vec{d}(w)$ ultimately represents the word $w$ in concept space and the semantic relatedness of two words can be calculated as the cosine of their concept vectors.

We computed the semantic relatedness of the lemmatized candidate term in the corpus to each of the five cluster key concepts. In doing so, we had to disambiguate the originally selected "*production*" to "*film production*". A limitation of the ESA algorithm available to us is, that it only calculates the similarity between two single words and not multiword expressions or phrases. We had to therefore resort to the spelling variant "*film-production*", which can also be found in Wikipedia. Each movie aspect identified was then mapped to the cluster to which it had the highest semantic relatedness score. For each of the resulting five movie aspect clusters, we only retained the 20 highest-ranked terms for our experiments. Zhuang et al. (2006) report that their movie feature classes mostly contained less than 20 words. This is also the case for our manually created clusters and suggests that a cluster size of 20 seems reasonable. For reasons of space, Table 6.2 shows only the top 10 terms. We observe that for four of the five clusters the aspects created by the manual and the ESA approach are very similar. In total, there is an overlap of 16 movie aspects between the ESA and the manually created clusters, with the "*production*" / "*film-production*" clusters having zero overlap. This might be due to the fact that the concept "*film-production*" does not occur in Wikipedia as often as the other four concepts. Therefore, the ESA algorithm also rates terms which are specific to those fewer articles as semantically highly related. This is probably also the reason why the name "Asheville" (a city) is considered to be so highly related.

### Fully Automatic Clustering

In this approach, we completely eliminate the manual effort in both the identification of key concepts in the movie domain and the movie aspect clustering. Since it allows to control the number of clusters produced and since it has been successfully applied to several tasks in the past, we decided to employ Latent Dirichlet Allocation (Blei et al., 2003) for the clustering. We again removed all words in our opinion expression lexicon (see Section 6.3.2) from the corpus before clustering it. We then employed the Mallet toolkit[2] to perform the clustering on our lemmatized corpus, using Mallet's built-in stop word filtering.

The clusters created by the LDA approach (Table 6.3) exhibit a much finer granularity regarding the represented concepts, but this was to be expected as the number of clusters is much higher. When analyzing the terms in the clusters, one can observe that the LDA approach models the domain on different levels: On the one hand, there are clusters which contain generic terms regarding the movie domain, while on the other hand there are also clusters which represent certain genres (horror, science-fiction, war) and even individual movies (James Bond, Hitchcock, Dracula). This outcome seems promising regarding the employment in the collaborative filtering, as such clusters could help to model the users' preferences on several

[2] `http://mallet.cs.umass.edu/`

Table 6.2: Top 10 Aspect Lemmas Clustered by ESA

| **acting** | **storyline** | **soundtrack** |
|---|---|---|
| acting | storyline | soundtrack |
| actor | storylines | song |
| role | character | release |
| actress | comic | music |
| filmography | reveal | album |
| co-star | series | track |
| act | story | feature |
| career | appear | band |
| television | universe | discography |
| theatre | villain | label |

| **cinematography** | **film-production** |
|---|---|
| cinematography | preproduction |
| runtime | asheville |
| distributor | contractees |
| budget | all-animated |
| min | high-living |
| film | cash-cow |
| edit | hit-and-miss |
| screenplay | star-driven |
| director | singer-actor |
| star | small-budget |

levels of granularity. As it is common for LDA outputs, the same term can appear in several clusters (e.g. “performance”, “film”). This requires a special strategy during the integration in the recommendation system (see Section 6.4).

### 6.3.2 Opinion Extraction

The extraction of opinions regarding individual movie aspects can be seen as an instance of opinion mining at the word/phrase level. On the basis of the movie aspect clusters described in Section 6.3.1, the remaining steps to be performed for each review are:

1. Identifying opinion-bearing words and potential movie aspects.
2. Linking opinion-bearing words to potential movie aspects.
3. Identifying the semantic orientation (polarity) of the opinions.
4. Aggregating all opinions for each movie aspect cluster.

Table 6.3: Top 10 Movie Aspect Lemmas Clustered by LDA

| **Cluster 1** | **Cluster 2** | **Cluster 3** | **Cluster 4** | **Cluster 5** | **Cluster 6** | **Cluster 7** | **Cluster 8** | **Cluster 9** | **Cluster 10** |
|---|---|---|---|---|---|---|---|---|---|
| woman | man | film | play | house | play | story | thriller | action | scene |
| life | police | make | character | gore | song | make | scene | scene | character |
| man | cop | time | role | remake | musical | king | plot | bond | sequel |
| wife | guy | watch | performance | night | music | tale | work | plot | make |
| father | drug | fact | actor | zombie | performance | vampire | end | sequence | series |
| daughter | town | story | story | dead | role | set | hitchcock | guy | humor |
| find | city | character | time | budget | cast | film | suspense | james | part |
| son | western | feel | give | director | woman | dr | room | die | tv |
| husband | john | lot | tom | make | stage | version | find | car | michael |
| young | gang | show | make | genre | screen | dracula | director | chase | plot |
| **Cluster 11** | **Cluster 12** | **Cluster 13** | **Cluster 14** | **Cluster 15** | **Cluster 16** | **Cluster 17** | **Cluster 18** | **Cluster 19** | **Cluster 20** |
| life | girl | good | character | cast | movie | thing | effect | character | american |
| world | young | film | story | john | watch | end | human | work | man |
| human | child | oscar | book | dvd | make | people | alien | style | make |
| experience | kid | star | make | make | time | time | earth | director | world |
| present | family | year | man | play | lot | happen | space | audience | show |
| people | year | win | performance | work | people | start | world | visual | country |
| reality | boy | picture | comic | direct | story | feel | fi | story | people |
| story | school | number | give | director | thing | scene | sci | camera | war |
| mind | high | director | actor | role | scene | make | back | filmmaker | america |
| society | mother | actor | time | ben | expect | point | crew | art | soldier |

We perform sentence splitting, tokenization, part-of-speech tagging and lemmatization[3], and then identify the movie aspects and the opinion bearing words in each review. For the latter task, we use the subjectivity clue lexicon from Wilson et al. (2005).

In contrast to documents from other online sources of user-generated content, the reviews collected from the IMDB exhibit a rather high quality. Proper capitalization, correct grammar and a rather small number of spelling errors were evident for most of the documents inspected. We observed that the users write in a rather elaborate style, which often results in long sentences with nested clauses etc. This level of sentence complexity in the reviews rules out the use of shallow pattern-matching surface methods for linking an identified opinion expression to its target. Such methods based on e.g. word distance (Hu and Liu, 2004a) or part-of-speech patterns (Yi et al., 2003) have been used in the past. These methods have the advantage of being computationally very efficient. However, the use of syntactic parsers, while computationally more expensive, can yield more accurate structural analyses (Zhuang et al., 2006; Kessler and Nicolov, 2009), which is of particular importance for more complex analyses such as negation detection. Our approach is therefore based on the syntactic analysis of the review sentences.

We employ the Stanford Parser[4], which extracts typed dependencies from the grammatical relations in a sentence. In contrast to the work in (Zhuang et al., 2006), our approach does not require a training phase for learning relevant constituents and their syntactic relations. Instead, the extraction of movie aspects with their corresponding opinions is done on the basis of two *generic dependency relation patterns* which are visualized in Figure 6.3:

The first pattern makes use of the fact that adjectives are the major means of expressing positive or negative opinions. Adjectives also make up the largest single fraction (48%) of the subjectivity clues in the Wilson lexicon. Accordingly, we found the majority of dependency relations between opinion expressions and movie aspects in our corpus to be adjectival modifiers (AMOD) as in "*the beautiful soundtrack*" and nominal subjects (NSUBJ) as in "*the soundtrack is beautiful*". During the analysis of the data learned by the supervised approaches in (Zhuang et al., 2006) and (Kessler and Nicolov, 2009), the authors also observe that direct dependency relations are the most frequent and at the same time the most reliable indicators of related opinion targets and opinion expressions. Such direct dependency relations are therefore used by us to extract a movie aspect with the corresponding opinion.

However, there are quite a few sentence constructions in which the relation between the opinion expression and the movie aspect is established over an intermediate word. Consider the sentence: "*This is acting at its most laconic form.*" in which the word "*form*" establishes the link between the movie aspect "*acting*" (PREP) and the opinion word "*laconic*' (AMOD). Our second pattern captures these connections involving intermediate words. It also enables us to extract opinions on both aspects from sentences as "*The entire score and the atmosphere are awesome.*" in which the parser will identify the relation between "*score*" and "*awesome*" (NSUBJ) as well as the conjunction between "*score*" and "*atmosphere*" (CONJ). We can thus

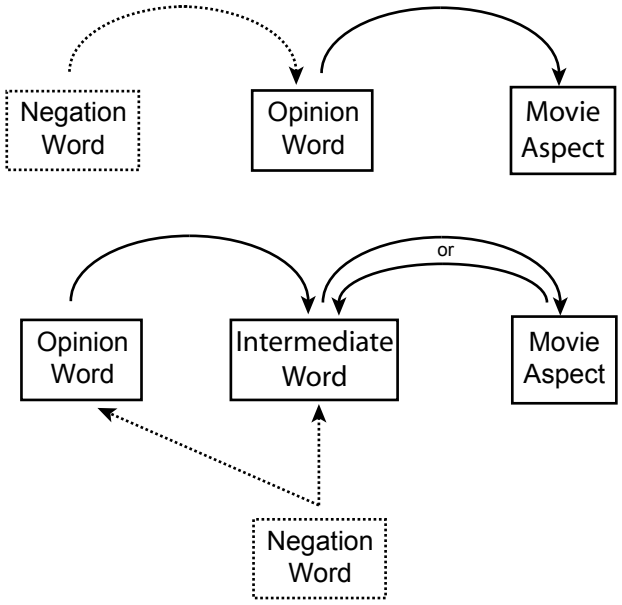
[3]Again, the TreeTagger (Schmid, 1994) was employed for part-of-speech tagging and lemmatization.

[4]`http://nlp.stanford.edu/software/lex-parser.shtml`

extract the opinion regarding each of the two movie aspects. This simultaneous extraction also works for two opinions being expressed for one aspect, such as in "*The characters are unbelievable and flat*".

The task of detecting negation during the opinion extraction is also done by analyzing the dependency parser output. If we find a direct negation relation to an opinion expression, we invert the polarity, i.e. the positive or negative orientation, of that opinion. In the case of a relation with an intermediate word, we check for a negation relation to the opinion expression or the intermediate word. Figure 6.3 illustrates the possible dependency relation paths which our approach uses to extract pairs of opinion expressions and movie aspects, and to do the negation detection.

Figure 6.3: Possible Dependency Relations for Opinion Extraction




## 6.4 Experiments and Results

### 6.4.1 Dataset

Although several datasets for the evaluation of recommendation systems are available (e.g. MovieLens[5], Netflix, BookCrossing[6], Jester Joke[7]), they only provide numerical or star ratings and no additional free-text reviews. Since we are interested in the effect of the opinions extracted from free-text reviews on recommendations, we had to create our own data set. We extracted a raw data set containing the ratings and corresponding reviews of approx. 1000 random users from the Internet Movie Database (IMDB)[8]. In the IMDB, each rating is on the scale from one to ten stars,

[5] http://www.grouplens.org/node/73
[6] http://www.informatik.uni-freiburg.de/~cziegler/BX/
[7] http://www.ieor.berkeley.edu/~goldberg/jester-data/
[8] http://www.imdb.com

and according to IMDB policy, free-text reviews must have at least ten lines and at most 1000 words. The IMDB website recommends a length of 200 to 500 words. As it is often the case in fan communities, some users contribute only a few reviews, while others contribute a lot. The same applies for the movies - some are rated by many users, some only by a few. In order to enhance the collaborative effect, we removed all reviews regarding movies with less than ten reviews from the raw dataset. Some statistics on the raw and the reduced dataset are given in Table 6.4. The removal of the seldomly rated movies also drastically reduced the percentage of users with only a few reviews: In the raw dataset, as many as 52% of the users wrote less than five reviews, while in the reduced dataset this share decreases to only 11%.

Table 6.4: Dataset Statistics

| | **Raw** | **Reduced** |
|---|---|---|
| Reviews | 136710 | 53112 |
| Sentences | 1907670 | 805937 |
| Avg. Sentences per Review | 13.9 | 15.2 |
| Avg. Tokens per Sentence | 24.2 | 23.6 |
| Reviewed Movies | 41288 | 2731 |
| Users | 1030 | 509 |
| Users with $< 5$ reviews | 541 | 57 |

Our experiments examine the usefulness of extracting opinions about movie aspects and using them as additional features for recommendations based on collaborative filtering. Our recommendation system will predict the overall star rating of movies for given users. We perform a ten-fold cross validation on the dataset of ratings and reviews. For each testing fold, we calculate the root mean square error (RMSE) between the actual star ratings and our predictions, which is in turn averaged over the ten folds. We extrinsically evaluate the three different clustering approaches described in Section 6.3.1. The computational complexity for calculating our models ranges from two seconds to three minutes per fold depending on the number of contributing entities and relations.

### 6.4.2 Experimental Setup

Our results are created by always using the overall star ratings of the users regarding the movies as the most basic feature. Since we also want to evaluate the usefulness of the extracted opinions against other features typically used for collaborative filtering in the movie domain, we created an extended baseline which also uses the genre information of the rated movie. The genre information was extracted from the IMDB as well. Note that the IMDB allows a movie to belong to more than one genre.

In our first non-baseline configuration (`⋆-Rating + Opinion Ratings`), we extract and accumulate all expressions for each movie aspect cluster from each review

and average the identified opinion polarities (positive or negative orientation) in order to extract one overall polarity value for each cluster. We noticed, however, that this approach loses some relevant information, i.e. the *number* of opinions uttered about that cluster. As described in the beginning of this Chapter, this information could be useful for the collaborative filtering, since it can reveal how important a certain movie aspect cluster is for a user. In our second set of non-baseline experiments (⋆`-Rating + Opinion Ratings + Number Opinions`), we therefore also incorporated this number.

For our three approaches which extract individual opinions from the reviews, we have three configurations each: The first configuration includes the ⋆-Rating and the opinion ratings for the 5 (Manual), 5 (ESA) and 20 (LDA) clusters. In the second configuration, we then add the genre information, and in the third one, we also include the number of opinions extracted for each cluster.

Table 6.5: Results of ⋆-Rating Prediction (smaller RMSE is better)

| **Setup** | **Features** | **RMSE (95%CI)** |
|---|---|---|
| Baseline | ⋆-Rating | $1.8526^{+0.0060}_{-0.0060}$ |
| | + Genre | $1.8319^{+0.0058}_{-0.0058}$ |
| Manual | ⋆-Rating + Opinion Rating | $1.8225^{+0.0064}_{-0.0073}$ |
| | + Number Opinions | $1.8221^{+0.0060}_{-0.0061}$ |
| | + Genre | $1.8090^{+0.0069}_{-0.0068}$ |
| ESA | ⋆-Rating + Opinion Rating | $1.8269^{+0.0065}_{-0.0062}$ |
| | + Number Opinions | $1.8243^{+0.0063}_{-0.0069}$ |
| | + Genre | $1.8080^{+0.0063}_{-0.0064}$ |
| LDA | ⋆-Rating + Opinion Rating | $1.8230^{+0.0072}_{-0.0072}$ |
| | + Number Opinions | $1.8139^{+0.0066}_{-0.0072}$ |
| | + Genre | $1.8073^{+0.0073}_{-0.0080}$ |

### 6.4.3 Discussion

The results of our experiments are shown in Table 6.5. For each setup, the contributing features and the RMSE along with the corresponding 95% confidence intervals are given. The confidence intervals are used to indicate the statistical significance of the RMSE changes. The first two rows contain the results of our baseline configuration in which we use the star ratings as a feature or both the star rating and the genre information. We observe that incorporating the genre information significantly reduces the RMSE. This was to be expected, as the genre information has been successfully employed in previous research. When analyzing the results of the three configurations which use the ratings extracted from the opinions (five clusters

for "Manual" and "ESA", 20 for "LDA"), we observe that this additional feature reduces the RMSE in all approaches. When comparing the results of star rating plus genre information as features with the star rating plus opinion rating as features, we observe that the predictions when using the extracted opinion ratings are always better regardless of which clustering approach is used.

When comparing the setups based on opinion mining regarding pairs of identical features, we observe that the results of the ESA-based approach are always slightly worse than the approach based on the manual clustering. Apparently, the slightly bigger clusters of the ESA approach and the fact that terms in the clusters definitely occur in the corpus as well compensate for the lack of detecting opinions about artists or directors. The ESA approach seems to be a reasonable option if the cluster topics can be defined manually, but the effort of filling the clusters by hand as well is not desired. The LDA-based approach performs slightly worse than the manual approach when using the `⋆-Rating + Opinion Ratings` features. However, when including the number of opinions and the genre information, it is consistently better than the other configurations and significantly better than the baseline. Ultimately, the information regarding the number of extracted opinions is beneficial regarding the predictions in all configurations. Apparently, this feature introduces relevant information which allows the collaborative filtering to e.g. model how important a certain aspect cluster is to a user.

In our last experimental setup, we wanted to verify whether the features extracted by the opinion mining are complementary or redundant in combination with the genre information. As shown in the last row of each clustering-based approach, the results improve in all configurations when combining the opinion ratings with the genre information. We can therefore conclude that the opinions extracted about the movie aspects are a useful feature to improve the recommendations of the collaborative filtering. The confidence intervals indicate that all improvements with respect to the `⋆-Rating + Genre` baseline are statistically significant with at least $p < 0.05$.

Most important for the users' acceptance of the recommendation system is the proper prediction of the items the user is most interested in, maximizing true positives and avoiding false positives. Recommendation systems can be seen as supervised classifiers mapping the input features to two classes: *likes* and *dislikes*. In order to evaluate our models with respect to this consideration, we re-interpreted all given ratings: ratings smaller than the global average (6.997) were labeled as *dislikes* and ratings above were labeled accordingly as *likes*. Now we can compare the models in terms of receiver operating characteristics (ROC) as well as the area under the ROC curve (AUC) (Fawcett, 2003). We calculated the AUC values for the LDA approach with all features as it yielded the best results regarding RMSE. The `⋆-Rating + Genre` baseline is improved by approximately 1.18%: $AUC_{LDA} = 0.9072 > 0.8967 = AUC_{baseline}$ (higher AUC is better). In order to assess the impact of the improvements regarding RMSE on the actual user-experience, one has to conduct a user study. After all, the users of a recommendation system have to decide whether they enjoy the movies recommended to them or not. The participants and organizers of the NetFlix challenge have conducted such a study and found that " a 1% improvement of the RMSE can make a big positive difference

in the identity of the "top-10" most recommended movies for a user"[9]. We therefore believe that the improvements we reach regarding the recommendations by including our opinion mining-based features are relevant in a real-world setting.

## 6.5 Chapter Summary

In this Chapter, we have shown how the extraction of opinions from free-text movie reviews can be used as features for a recommendation system to improve the prediction accuracy. The information extracted from the users' opinions can be employed in combination with structured information about the movies which in turn leads to better results. Our results show that the LDA-based movie aspect extraction and clustering approach yields the best results while the candidate extraction and clustering work fully automatic. We see the main difference between the LDA-based and the other two approaches in the number and the granularity of the clusters extracted. We can conclude that the larger number and fine-grained clusters provide a broader i.e. a better representation of the topics in the corpus and are therefore beneficial for the recommendation accuracy. However, in future work we might investigate whether a disambiguation of movie aspects that occur in more than one LDA cluster can lead to even better results, since the opinion extraction would be more exact then. The results we obtained with the ESA-based approach are promising, but we observed that the ability to only calculate the semantic similarity between single word terms limits the detection of e.g. actors as semantically related to the "acting" category. If we can overcome this limitation, we could improve the detection of opinions regarding some categories, which could in turn lead to better recommendations.

The elaborate style of the majority of the reviews, in combination with complex sentence structures lead to a frequent use of anaphora in the documents. By resolving the anaphora, as shown in Chapter 5, we might increase the recall of the extracted opinions. The computational power which we require in order to process the 53000 documents is however already considerable with our current experimental setup. In order to retain a manageable complexity we decided not to include the anaphora resolution components in our experiments so far, but we are confident that by tuning all components regarding speed and efficiency, we could extend the complexity of the preprocessing required e.g. for the anaphora resolution.

The representation of the opinions for the collaborative filtering might be improved by analyzing the positive and the negative opinions separately: In our current setup the recommendation system only receives the overall number of opinions regarding a certain aspect cluster. The differentiation between positive and negative opinions should lead to a more exact representation of the review.

[9] `http://www.netflixprize.com/community/viewtopic.php?id=828`

# Chapter 7

# Summary

The extraction of opinion targets is an integral task for an opinion mining system which operates on a word / phrase level, since the opinion targets reflect what the opinion in a sentence is about. In this thesis, we conducted a comprehensive study on the extraction of opinion targets by analyzing both unsupervised and supervised approaches as well as additional challenges as the extraction of anaphoric opinion targets. We investigated how two unsupervised algorithms perform in the task of opinion target extraction on datasets spanning four different domains. We evaluated the algorithms while also analyzing the influence of the foregoing identification of opinion expressions. Our results show that the Likelihood Ratio Test-based (LRT) approach yields the best results in the opinion target extraction task, given that sentences containing an opinion have been identified with perfect accuracy. We showed that a *nearest noun phrase* (NNP) heuristic outperforms both the LRT-based approach as well as the Association Mining-based (AM) approach on all four datasets if the opinion expressions have been identified with perfect accuracy. Our error analysis indicates that both the LRT-based approach as well as the AM-based approach, which solely rank opinion target candidates based on their corpus frequency, are not well suited for the task because the opinion target frequencies follow a Zipfian distribution on all our datasets. This entails that there is a large number of opinion targets which only occur very seldomly. Furthermore, our error analysis shows that these two approaches yield a relatively low precision since words or phrases, which are ranked as being relevant and hence extracted as opinion targets, also frequently occur in sentences in which they are not the actual targets. Since neither of the algorithms consider e.g. grammatical relations to the opinion expressions, they will simply extract all occurrences of a highly ranked opinion target candidate.

Furthermore, we evaluated two supervised algorithms for opinion target extraction: One is the algorithm by Zhuang et al. (2006), which represents the state-of-the-art in supervised opinion target extraction on movie reviews, which is one of the four datasets we employ. The other one is a Conditional Random Fields-based (CRF) approach which we introduce for the extraction of opinion targets. The CRF-based approach outperforms the algorithm by Zhuang et al. (2006) on all four datasets in the task of opinion target extraction. The CRF-based approach also outperforms the best performing unsupervised algorithm, which is the NNP heuristic, in the opinion target extraction task on all four datasets. Since supervised approaches frequently exhibit the problem that the models only perform well on

data from the domain which they have been trained on, we also evaluate the CRF-based approach as well as the algorithm by Zhuang et al. (2006) in a cross-domain opinion target extraction setting. Our experiments show that the CRF-based approach yields significantly better results in the cross-domain setting, and that the features which we employ scale well across domains. We also conducted a comparative evaluation of the two supervised approaches against the best performing unsupervised approach in the cross-domain setting. Our CRF-based approach outperforms the NNP heuristic on three of the four datasets. While we reach significant improvements over the state-of-the-art with the CRF-based approach, our results show, that both in the single- and cross-domain setting there is quite some room for improvement. The inter-annotator agreement for the annotation of opinion targets on the "web-services" and "cars" dataset is 0.80 regarding F-Measure, while with the best performing CRF configuration we reach an F-Measure of 0.46 and 0.50 on these two datasets.

We can conclude that in general the effort of creating labeled data for the training of the CRF-based approach always pays off: In the single-domain setting the CRF-based approach outperforms both the supervised baseline by Zhuang et al. (2006) and the best unsupervised approach on all datasets. In the cross-domain setting, the CRF-based approach reaches between 60% and 96% of the performance regarding F-Measure which it yields in the single-domain setting.

The extraction of anaphoric opinion targets has not been addressed in previous research although they do make up a significant number of opinion targets in the datasets which we employ. We therefore investigated the extraction of anaphoric opinion targets. In doing so, we required our opinion target extraction algorithms to correctly identify the antecedent of an anaphoric opinion target. We integrated the two anaphora resolution algorithms MARS and CogNIAC into a state-of-the-art opinion mining algorithm. By extending the CogNIAC algorithm, which was originally designed for high-precision anaphora resolution, we reached significant improvements regarding the opinion target extraction. We have shown how to successfully integrate an anaphora resolution algorithm into an opinion mining system. This step enables the extraction of opinion targets across sentence boundaries and thereby offers new possibilities, e.g. towards an (Opinion) Question Answering system, in which information about the antecedents of anaphoric opinion targets is required.

Finally, we showed how an opinion mining algorithm can be integrated with a recommendation system. We presented three different approaches for the identification and integration of the opinion extraction and performed an extrinsic evaluation of each approach in a movie recommendation setting. Our results show that each of the three approaches significantly improves the movie recommendation accuracy. The LDA-based approach which requires no manual effort even yields the best results, which shows that the integration of an opinion mining algorithm is definitely viable in a real-world scenario. By extracting additional information from the opinions which the users have uttered in their free-text reviews we managed to improve the recommendations of the system. This should lead to an improved user experience as the recommendations are more accurately adjusted to a user's taste.

## Future Work

In future work regarding the supervised extraction of opinion targets, we want to investigate how machine learning algorithms, which are specifically designed for the problem of domain adaptation (Blitzer et al., 2007; Jiang and Zhai, 2007), perform in comparison to our approach, since good results have been reached with these approaches in the related task of cross-domain sentiment classification.

Our results have confirmed that short paths in the dependency graphs are good indicators for related opinion expressions and opinion targets, also observed in (Zhuang et al., 2006) and (Kessler and Nicolov, 2009). As we showed in our experiments, there are however several hundred different dependency - pos paths for each of the datasets. These patterns could also be investigated in future research by analyzing the dependency graphs for recurring sub-graphs, which might be stable across domains.

Since three of the features we employed in our CRF-based approach are based on previously identified opinion expressions, it is to investigate how to mitigate the possible negative effects introduced by errors in the opinion expression identification if they are not annotated in the gold standard. Our results have shown that a lexicon-based approach considerably decreases the opinion target extraction performance. We plan to investigate whether state-of-the-art approaches for the opinion expression identification (Breck et al., 2007; Johansson and Moschitti, 2010) can yield better results in future work.

The work by Charniak and Elsner (2009) has shown that their Expectation Maximization-based approach for anaphora resolution significantly outperforms knowledge-poor algorithms, as we employ them in our experiments, on newswire. We plan to investigate whether their algorithm can also be applied to user-generated discourse which we focused on in this thesis.

Motivated by the good results we reached with the fully automatic clustering and target extraction for the recommendation system, a next step could be to investigate other automatic approaches. Our LDA clustering operates only on single words as cluster content, but a co-occurrence analysis might lead to a richer and therefore more meaningful representation of the documents by being able to extract complete phrases.

# Bibliography


Aciar, S., Zhang, D., Simoff, S., and Debenham, J. (2007). Informed recommender: Basing recommendations on consumer product reviews. *IEEE Intelligent Systems*, 22(3):39–47.

Agrawal, R. and Srikant, R. (1994). Fast algorithms for mining association rules. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, Santiago de Chile, Chile.

Andreevskaia, A. and Bergler, S. (2006). Mining wordnet for a fuzzy sentiment: Sentiment tag extraction from wordnet glosses. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 209–216, Trento, Italy.

Aone, C. and Ramos-Santacruz, M. (2000). Rees: a large-scale relation and event extraction system. In *Proceedings of the Sixth Conference on Applied Natural Language Conferences*, pages 76–83, Stroudsburg, PA, USA.

Aue, A. and Gamon, M. (2005). Customizing sentiment classifiers to new domains: A case study. In *Proceedings of the 5th International Conference on Recent Advances in Natural Language Processing*, Borovets, Bulgaria.

Baker, C. F. and Sato, H. (2003). The framenet data and software. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2*, pages 161–164, Sapporo, Japan.

Baldwin, B. (1997). Cogniac: High Precision Coreference with Limited Knowledge and Linguistic Resources. In *Proceedings of a Workshop on Operational Factors in Practical, Robust Anaphora Resolution for Unrestricted Texts*, pages 38–45, Madrid, Spain.

Bethard, S., Yu, H., Thornton, A., Hatzivassiloglou, V., and Jurafsky, D. (2004). Automatic extraction of opinion propositions and their holders. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, pages 22–24, Stanford, California, USA.

Blair-Goldensohn, S., Hannan, K., McDonald, R., Neylon, T., Reis, G., and Reynar, J. (2008). Building a sentiment summarizer for local service reviews. In *Proceedings of the WWW2008 Workshop: NLP in the Information Explosion Era (NLPIX 2008)*, Beijing, China.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Blitzer, J., Dredze, M., and Pereira, F. (2007). Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 440–447, Prague, Czech Republic.

Bloom, K., Garg, N., and Argamon, S. (2007). Extracting appraisal expressions. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 308–315, Rochester, New York, USA.

Bourigault, D. and Jacquemin, C. (1999). Term extraction / term clustering: An integrated platform for computer-aided terminology. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics*, pages 15–22, Bergen, Norway.

Breck, E., Choi, Y., and Cardie, C. (2007). Identifying expressions of opinion in context. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2683–2688, Hyderabad, India.

Breese, J. S., Heckerman, D., and Kadie, C. (1998). Empirical analysis of predictive algorithms for collaborative filtering. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, pages 43–52, Madison, Wisconsin, USA.

Bruce, R. F. and Wiebe, J. M. (1999). Recognizing subjectivity: A case study of manual tagging. *Natural Language Engineering*, 5:187–205.

Carenini, G., Ng, R. T., and Zwart, E. (2005). Extracting knowledge from evaluative text. In *Proceedings of the 3rd International Conference on Knowledge Capture (K-CAP 2005)*, pages 11–18, Banff, Canada.

Charniak, E. and Elsner, M. (2009). EM works for pronoun anaphora resolution. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 148–156, Athens, Greece.

Cheng, X. and Xu, F. (2008). Fine-grained opinion topic and polarity identification. In *Proceedings of the 6th International Conference on Language Resources and Evaluation*, pages 2710–2714, Marrekech, Morocco.

Chesley, P., Vincent, B., Xu, L., and Srihari, R. (2006). Using verbs and adjectives to automatically classify blog sentiment. In *Proceedings of AAAI-CAAW-06, the Spring Symposia on Computational Approaches to Analyzing Weblogs*, Stanford, California, USA.

Choi, Y., Breck, E., and Cardie, C. (2006). Joint extraction of entities and relations for opinion recognition. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 431–439, Sydney, Australia.

Choi, Y. and Cardie, C. (2009). Adapting a polarity lexicon using integer linear programming for domain-specific sentiment classification. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 590–598, Singapore. Association for Computational Linguistics.

Choi, Y., Cardie, C., Riloff, E., and Patwardhan, S. (2005). Identifying sources of opinions with conditional random fields and extraction patterns. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Vancouver, Canada.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20(1):37–46.

Cowie, J. R. and Lehnert, W. G. (1996). Information extraction. *Communications of the ACM*, 39(1):80–91.

Daumé III, H. and Marcu, D. (2006). Domain adaptation for statistical classifiers. *Journal of Artificial Intelligence Research (JAIR)*, 26:101–126.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th International World Wide Web Conference (WWW-03)*, pages 519–528, Budapest, Hungary.

Dunning, T. (1993). Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

Eguchi, K. and Lavrenko, V. (2006). Sentiment retrieval using generative models. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 345–354, Sydney, Australia.

Esuli, A. and Sebastiani, F. (2006). SentiWordNet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, pages 417–422, Genova, Italy.

Fahrni, A. and Klenner, M. (2008). Old wine or warm beer: Target-specific sentiment analysis of adjectives. In *Proceedings of the Symposium on Affective Language in Human and Machine, AISB 2008 Convention*, pages 60 – 63, Aberdeen, Scotland.

Fawcett, T. (2003). Roc graphs: Notes and practical considerations for researchers. Technical Report HPL-2003-4, HP Laboratories Palo Alto.

Feiguina, O. and Lapalme, G. (2007). Query-based summarization of customer reviews. In *Proceedings of the 20th Conference of the Canadian Society for Computational Studies of Intelligence*, pages 452–463, Montreal, Canada.

Fellbaum, C., editor (1998). *WordNet: An Electornic Lexical Database*. MIT Press, Cambridge, Massachusetts, USA.

Ferraresi, A., Zanchetta, E., Bernardini, S., and Baroni, M. (2008). Introducing and evaluating ukwac, a very large web-derived corpus of english. In *Proceedings of the 4th Web as Corpus Workshop (WAC-4) Can we beat Google?*, pages 47–54, Marrakech, Morocco.

Ferreira, L., Jakob, N., and Gurevych, I. (2008). A comparative study of feature extraction algorithms in customer reviews. In *Proceedings of the 2nd IEEE International Conference on Semantic Computing*, pages 144–151, Santa Clara, California, USA.

Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 363–370, Ann Arbor, Michigan, USA.

Gabrilovich, E. and Markovitch, S. (2007). Computing semantic relatedness using wikipedia-based explicit semantic analysis. In *Proceedings of the 20th International Joint Conference for Artificial Intelligence*, pages 1606–1611, Hyderabad, India.

Gamon, M. (2004). Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 841–847, Geneva, Switzerland.

Gamon, M., Aue, A., Corston-Oliver, S., and Ringger, E. (2005). Pulse: Mining customer opinions from free text. In *Advances in Intelligent Data Analysis VI*, volume 3646 of *Lecture Notes in Computer Science*, pages 121–132. Springer Berlin.

Ghose, A. and Ipeirotis, P. G. (2007). Designing novel review ranking systems: Predicting usefulness and impact of reviews. In *Proceedings of the Ninth International Conference on Electronic Commerce*, pages 303–310, Minneapolis, Minnesota, USA. Invited paper.

Ghose, A., Ipeirotis, P. G., and Sundararajan, A. (2007). Opinion mining using econometrics: A case study on reputation systems. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 416–423, Prague, Czech Republic.

Godbole, N., Srinivasaiah, M., and Skiena, S. (2007). Large-scale sentiment analysis for news and blogs. In *Proceedings of the International Conference on Weblogs and Social Media*, Boulder, Colorado, USA.

Goldberg, A. and Zhu, X. (2006). Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *Proceedings of TextGraphs: the First Workshop on Graph Based Methods for Natural Language Processing*, pages 45–52, New York City, New York, USA.

Gundel, J., Hedberg, N., and Zacharski, R. (1993). Cognitive status and the form of referring expressions in discourse. *Language*, 69:274–307.

Haghighi, A. and Klein, D. (2007). Unsupervised coreference resolution in a nonparametric bayesian model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 848–855, Prague, Czech Republic.

Herlocker, J. L., Konstan, J. A., Terveen, L. G., John, and Riedl, T. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22:5–53.

Hofmann, T. (1999). Probabilistic latent semantic analysis. In *Proceedings of the Proceedings of the Fifteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 289–297, Stockholm, Sweden.

Holzinger, W., Krüpl, B., and Herzog, M. (2006). Using ontologies for extracting product features from web pages. In *Proceedings of the 5th International Semantic Web Conference, ISWC 2006*, pages 286–299, Athens, Georgia, USA.

Hu, M. and Liu, B. (2004a). Mining and summarizing customer reviews. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 168–177, Seattle, Washington, USA.

Hu, M. and Liu, B. (2004b). Mining opinion features in customer reviews. In *Proceedings of the 19th AAAI Conference on Artificial Intelligence*, pages 755–760, San Jose, California, USA.

Hurst, M. F. and Nigam, K. (2004). Retrieving topical sentiments from online document collections. In *Document Recognition and Retrieval XI*, pages 27–34, San Jose, California, USA.

Jakob, N. and Gurevych, I. (2010a). Extracting opinion targets in a single- and cross-domain setting with conditional random fields. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1035–1045, Cambridge, Massachusetts, USA.

Jakob, N. and Gurevych, I. (2010b). Using anaphora resolution to improve opinion target identification in movie reviews. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 263–268, Uppsala, Sweden.

Jakob, N., Müller, M.-C., and Gurevych, I. (2009a). LRTwiki: Enriching the likelihood ratio test with encyclopedic information for the extraction of relevant terms. In *Proceedings of the IJCAI 2009 Workshop on Wikipedia and Artificial Intelligence (WikiAI 2009)*, pages 3–8, Pasadena, California, USA.

Jakob, N., Weber, S. H., Müller, M.-C., and Gurevych, I. (2009b). Beyond the stars: Exploiting free-text user reviews for improving the accuracy of movie recommendations. In *Proceedings of the 1st International CIKM Workshop on Topic-Sentiment Analysis for Mass Opinion Measurement*, pages 57–64, Hong Kong.

Jiang, J. and Zhai, C. (2007). Instance weighting for domain adaptation in nlp. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 264–271, Prague, Czech Republic. Association for Computational Linguistics.

Jijkoun, V., de Rijke, M., and Weerkamp, W. (2010). Generating focused topic-specific sentiment lexicons. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 585–594, Uppsala, Sweden. Association for Computational Linguistics.

Johansson, R. and Moschitti, A. (2010). Syntactic and semantic structure for opinion expression detection. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 67–76, Stroudsburg, PA, USA.

Kamps, J., Marx, M., Mokken, R. J., and de Rijke, M. (2004). Using WordNet to measure semantic orientation of adjectives. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 1115–1118, Lisboa, Portugal.

Kanayama, H. and Nasukawa, T. (2006). Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363, Sydney, Australia.

Kessler, J. and Nicolov, N. (2009). Targeting sentiment expressions through supervised ranking of linguistic configurations. In *Proceedings of the Third International AAAI Conference on Weblogs and Social Media*, pages 90–97, San Jose, California, USA.

Kessler, J. S., Eckert, M., Clark, L., and Nicolov, N. (2010). The 2010 icwsm jdpa sentment corpus for the automotive domain. In *4th International AAAI Conference on Weblogs and Social Media Data Workshop Challenge (ICWSM-DWC 2010)*, Washington, DC, USA.

Kim, S.-M. and Hovy, E. (2006). Extracting opinions, opinion holders, and topics expressed in online news media text. In *Proceedings of the ACL Workshop on Sentiment and Subjectivity in Text*, pages 1–8, Sydney, Australia.

Kobayashi, N., Iida, R., Inui, K., and Matsumoto, Y. (2006). Opinion mining as extraction of attribute-value relations. In *Proceedings of New Frontiers in Artificial Intelligence, Joint JSAI 2005 Workshop Post-Proceedings*, volume 4012 of *Lecture Notes in Computer Science*, pages 470–481. Springer.

Kobayashi, N., Inui, K., Matsumoto, Y., Tateishi, K., and Fukushima, T. (2004). Collecting evaluative expressions for opinion extraction. In *Proceedings of the 1st International Joint Conference on Natural Language Processing*, pages 596–605, Hainan Island, China.

Lafferty, J. D., McCallum, A., and Pereira, F. C. N. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning*, pages 282–289, Williamstown, Massachusetts, USA.

Li, S. and Zong, C. (2008). Multi-domain sentiment classification. In *Proceedings of ACL-08: HLT, Short Papers*, pages 257–260, Columbus, Ohio, USA. Association for Computational Linguistics.

Lippert, C., Weber, S.-H., Yi Huang, V. T., Schubert, M., and Kriegel, H.-P. (2008). Relation prediction in multi-relational domains using matrix factorization. In *NIPS 2008 Workshop on Structured Input Structured Output*, Vancouver, Canada.

Long, B., Wu, X., Zhang, Z. M., and Yu, P. S. (2006a). Unsupervised learning on k-partite graphs. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 317–326, Philadelphia, Pennsylvania, USA.

Long, B., Zhang, Z. M., Wu, X., and Yu, P. S. (2006b). Spectral clustering for multi-type relational data. In *Proceedings of the 23rd international conference on Machine learning*, pages 585–592, Pittsburgh, Pennsylvania, USA.

Lu, Y. and Zhai, C. (2008). Opinion integration through semi-supervised topic modeling. In *Proceedings of the 17th International World Wide Web Conference (WWW '08)*, pages 121–130, Beijing, China.

Marcus, M. P., Santorini, B., and Marcinkiewicz, M. A. (1993). Building a large annotated corpus of english: The penn treebank. *Computational Linguistics*, 19(2):313–330.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 188–191, Edmonton, Canada.

McDonald, R., Hannan, K., Neylon, T., Wells, M., and Reynar, J. (2007). Structured models for fine-to-coarse sentiment analysis. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 432–439, Prague, Czech Republic.

Mei, Q., Ling, X., Wondra, M., Su, H., and Zhai, C. (2007). Topic sentiment mixture: Modeling facets and opinions in weblogs. In *Proceedings of the 16th International World Wide Web Conference (WWW '07)*, pages 171–180, Banff, Canada.

Mihalcea, R., Banea, C., and Wiebe, J. (2007). Learning multilingual subjective language via cross-lingual projections. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 976–983, Prague, Czech Republic. Association for Computational Linguistics.

Mihalcea, R. and Tarau, P. (2004). Textrank: Bringing order into texts. In Lin, D. and Wu, D., editors, *Proceedings of the 9th Conference on Empirical Methods in Natural Language Processing*, pages 404–411, Barcelona, Spain.

Mitkov, R. (1998). Robust pronoun resolution with limited knowledge. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 869–875, Montreal, Canada.

Mitkov, R., Evans, R., and Orăsan, C. (2002). A new, fully automatic version of mitkov's knowledge-poor pronoun resolution method. In *Proceedings of the Third International Conference on Intelligent Text Processing and Computational Linguistics*, Mexico City, Mexico.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *Proceedings of the 9th Conference on Empirical Methods in Natural Language Processing*, pages 412–418, Barcelona, Spain.

Nasukawa, T. and Yi, J. (2003). Sentiment analysis: Capturing favorability using natural language processing. In *Proceedings of the 2nd International Conference on Knowledge Capture*, pages 70–77, Sanibel Island, Florida, USA.

Ng, V., Dasgupta, S., and Arifin, S. M. N. (2006). Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 611–618, Sydney, Australia.

Ni, X., Xue, G.-R., Ling, X., Yu, Y., and Yang, Q. (2007). Exploring in the weblog space by detecting informative and affective articles. In *Proceedings of the 16th International World Wide Web Conference*, Banff, Alberta, Canada.

Over, P. (2001). Introduction to duc-2001: an intrinsic evaluation of generic news text summarization systems. In *DUC 2001 Workshop on Text Summarization*, New Orleans, Louisiana, USA.

Pang, B. and Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 115–124, Ann Arbor, Michigan, USA.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the 7th Conference on Empirical Methods in Natural Language Processing*, pages 79–86, Philadelphia, Pennsylvania, USA.

Peng, F. and McCallum, A. (2006). Information extraction from research papers using conditional random fields. *Information Processing and Management*, 42(4):963–979.

Pinto, D., McCallum, A., Wei, X., and Croft, W. B. (2003). Table extraction using conditional random fields. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto, Canada.

Poesio, M. and Kabadjov, M. A. (2004). A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 663–666, Lisboa, Portugal.

Popescu, A.-M. and Etzioni, O. (2005). Extracting product features and opinions from reviews. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 339–346, Vancouver, Canada.

Qu, L., Toprak, C., Jakob, N., and Gurevych, I. (2008). Sentence level subjectivity and sentiment analysis experiments in ntcir-7 moat challenge. In *Proceedings of the 7th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retrieval, Question Answering, and Cross-Lingual Information Access*, pages 210–217, Tokyo, Japan.

Rabiner, L. and Juang, B. (1986). An introduction to hidden markov models. *IEEE ASSP Magazine*, 3(1):4 – 16.

Riloff, E. and Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 8th Conference on Empirical Methods in Natural Language Processing*, pages 105–112, Sapporo, Japan.

Salton, G. and McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill.

Sang, E. F. T. K. and Meulder, F. D. (2003). Introduction to the conll-2003 shared task: language-independent named entity recognition. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003 - Volume 4*, CONLL '03, pages 142–147, Stroudsburg, PA, USA.

Schafer, J. B., Frankowski, D., Herlocker, J. L., and Sen, S. (2007). Collaborative filtering recommender systems. In *The Adaptive Web*, volume 4321 of *Lecture Notes in Computer Science*, pages 291–324. Springer.

Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, pages 44–49, Manchester, UK.

Seki, Y., Evans, D. K., Ku, L.-W., Chen, H.-H., Kando, N., and Lin, C.-Y. (2007). Overview of opinion analysis pilot task at NTCIR-6. In *Proceedings of the Workshop Meeting of the National Institute of Informatics (NII) Test Collection for Information Retrieval Systems (NTCIR)*, pages 265–278, Tokyo, Japan.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 134–141, Edmonton, Canada.

Snyder, B. and Barzilay, R. (2007). Multiple aspect ranking using the Good Grief algorithm. In *Proceedings of Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 300–307, Rochester, New York, USA.

Somasundaran, S., Ruppenhofer, J., and Wiebe, J. (2008). Discourse level opinion relations: An annotation study. In *Proceedings of the 9th SIGdial Workshop on Discourse and Dialogue*, pages 129–137, Columbus, Ohio, USA.

Srebro, N., Rennie, J. D. M., and Jaakkola, T. S. (2005). Maximum margin matrix factorizations. In *Advances in Neural Information Processing Systems 17*, pages 1329–1336, Vancouver, Canada.

Steinberger, J., Kabadjov, M., Poesio, M., and Sanchez-Graillet, O. (2005). Improving lsa-based summarization with anaphora resolution. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Vancouver, Canada.

Stone, P. J., Dunphy, D. C., Smith, M. S., and Ogilvie, D. M. (1966). *The General Inquirer: A Computer Approach to Content Analysis*. MIT Press.

Stoyanov, V. and Cardie, C. (2008). Topic identification for fine-grained opinion analysis. In *Proceedings of the 22nd International Conference on Computational Linguistics*, pages 817–824, Manchester, UK.

Takacs, G., Pilaszy, I., Nemeth, B., and Tikk, D. (2007). On the gravity recommendation system. In *Proceedings of KDD Cup Workshop at the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 22–30, San Jose, California, USA.

Thomas, M., Pang, B., and Lee, L. (2006). Get out the vote: Determining support or opposition from Congressional floor-debate transcripts. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 327–335, Sydney, Australia.

Titov, I. and McDonald, R. (2008a). A joint model of text and aspect ratings for sentiment summarization. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 308–316, Columbus, Ohio, USA.

Titov, I. and McDonald, R. (2008b). Modeling online reviews with multi-grain topic models. In *Proceedings of the 17th International World Wide Web Conference (WWW '08)*, pages 111–120, Beijing, China.

Tomokiyo, T. and Hurst, M. (2003). A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment*, pages 33–40, Sapporo, Japan.

Toprak, C. and Gurevych, I. (2009). Document level subjectivity classification experiments in deft'09 challenge. In *Proceedings of the DEFT'09 Text Mining Challenge*, pages 89–97, Paris, France.

Toprak, C., Jakob, N., and Gurevych, I. (2010). Sentence and expression level annotation of opinions in user-generated discourse. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 575–584, Uppsala, Sweden.

Toutanova, K., Klein, D., Manning, C., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics*, pages 252–259, Edmonton, Canada.

Turney, P. D. (2000). Learning algorithms for keyphrase extraction. *Information Retrieval*, 2:303–336.

Turney, P. D. and Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems*, 21(4):315–346.

Vicedo, J. L. and Ferrández, A. (2000). Applying anaphora resolution to question answering and information retrieval systems. In *Proceedings of the First International Conference on Web-Age Information Management*, volume 1846 of *Lecture Notes In Computer Science*, pages 344–355. Springer, Shanghai, China.

Wan, X. and Xiao, J. (2008). Single document keyphrase extraction using neighborhood knowledge. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 855–860, Chicago, Illinois, USA.

Wermter, J. and Hahn, U. (2005). Finding new terminology in very large corpora. In *Proceedings of the 3rd International Conference on Knowledge Capture*, pages 137–144, New York, NY, USA.

Wiebe, J. and Riloff, E. (2005). Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Intelligent Text Processing and Computational Linguistics*, volume 3406 of *Lecture Notes in Computer Science*, pages 475–486, Mexico City, Mexico.

Wiebe, J. and Wilson, T. (2002). Learning to disambiguate potentially subjective expressions. In *Proceedings of the 6th Conference on Natural Language Learning 2002*, pages 112–118, Taipei, Taiwan.

Wiebe, J., Wilson, T., and Cardie, C. (2005). Annotating expressions of opinions and emotions in language. *Language Resources and Evaluation (formerly Computers and the Humanities)*, 39(2/3):164–210.

Wiebe, J. M., Bruce, R. F., and O'Hara, T. P. (1999). Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 246–253, College Park, Maryland, USA.

Wiebe, J. M., Wilson, T., Bruce, R., Bell, M., and Martin, M. (2004). Learning subjective language. *Computational Linguistics*, 30(3):277–308.

Wilcox, R. R. (2001). *Fundamentals of Modern Statistical Methods: Substantially Improving Power and Accuracy*. Springer.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 347–354, Vancouver, Canada.

Yang, X., Su, J., and Tan, C. L. (2006). Kernel-based pronoun resolution with structured syntactic knowledge. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 41–48, Sydney, Australia.

Yi, J., Nasukawa, T., Bunescu, R., and Niblack, W. (2003). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 427–434, Melbourne, Florida, USA.

Yu, H. and Hatzivassiloglou, V. (2003). Towards answering opinion questions: separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 129–136, Sapporo, Japan.

Yu, K., Yu, S., and Tresp, V. (2005). Multi-label informed latent semantic indexing. In *Proceedings of the 28th Annual International ACM Conference on Research and Development in Information Retrieval*, pages 258–265, Salvador, Brazil.

Zelenko, D., Aone, C., and Richardella, A. (2003). Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3:1083–1106.

Zhuang, L., Jing, F., and Zhu, X.-Y. (2006). Movie review mining and summarization. In *Proceedings of the ACM 15th Conference on Information and Knowledge Management*, pages 43–50, Arlington, Virginia, USA.

# Appendix A

# Sentiment Annotation in Reviews and Blogs[1]

Sentiment analysis and opinion mining are text classification and mining tasks which involve automatic analysis of subjective discourse and extraction of opinions, their targets and holders. Different facets of these challenging tasks include subjectivity analysis (Wiebe et al., 2004; Wiebe and Riloff, 2005) finding semantic orientation of words or phrases (Turney and Littman, 2003; Kamps et al., 2004; Wilson et al., 2005), mining individual opinions with their targets (Yi et al., 2003; Hu and Liu, 2004b; Jakob and Gurevych, 2010a) and holders (Kim and Hovy, 2006).

Supervised or unsupervised approaches to sentiment analysis demand reliably annotated data sets for implementation, evaluation and error analysis. In this annotation study, we aim to understand how opinions are expressed in English, especially in two specific user generated discourse variants: product reviews and political blogs. Corpus subject to our analysis consists of the reviews collected from the consumer review portal www.rateitall.com and blog post entries from controversial blogs about US presidential race.

Opinions and sentiments are private states (Wiebe et al., 2005), i.e., internal states that cannot be directly observed or objectively verified by others. Expressions and perceptions of opinions depend on the genre and the context of the text as well as reader's knowledge of the domain and cultural background. Considering these reasons it would be a very ambitious aim to capture all kinds of linguistic expressions and phenomena employed in natural language texts for conveying opinions and sentiments. Unlike the manual annotation study presented by Wiebe et al. (2005), which encounters a thorough analysis of subjectivity in newspaper articles, in this study, we focus on only the explicit expressions of opinions and the aspects which will be of interest to our immediate research. For instance, in the product reviews, one of our major interests is to find product features which were commented on, and then associate the correct sentiment towards a product, when the sentiment is expressed towards the product feature and the product name is not explicitly given in the sentence. Therefore, we let the annotators judge sentences in terms of their relevance towards a given topic.

[1]This chapter has been collaboratively created by Niklas Jakob, Cigdem Toprak and Iryna Gurevych

We propose a two-staged annotation process in order to reduce the complexity and establish a more consistent corpus of opinions. In the first stage, a sentence is the unit of analysis and annotators are given documents in which the sentence boundaries are already marked. They are asked to make decisions about the given sentences e.g. whether they are opinionated or not; relevant for the topic or not, etc. The second stage builds upon the results of the first stage and aims at a finer level of granularity for pinpointing and analyzing the opinion expressions. In the second stage annotators work on the sentences that they already processed and agreed upon as being opinionated in the first stage. This time they are asked to mark the word spans for the opinion expressions, the opinion targets and the holders. By designing this experiment in two stages we aim to ease the annotators' task, and to get them acquainted with the data before they start working on the more complex second stage. With this procedure we hope to increase the annotators' efficiency and the reliability of their decisions.

In Sections A.1 and A.2 we present the annotation guidelines for the first and the second stages respectively.

## A.1 Stage-1: Sentence level annotation process

The main purpose of the first stage is to judge each sentence in terms of its evaluative character and topic relevance. In other words, we try to answer the following three questions in this stage:

1. Is the sentence relevant for the given topic?
2. Does the sentence contain at least one opinion about the given topic?
3. Does the sentence evoke a positive or negative association about the given topic, even if there is no explicit expression of an opinion?

These three questions are captured by the three attributes, namely *topic_relevant*, *opinionated* and *polar_fact* in the annotation scheme which we will explain below. The unit of analysis is the sentence. Texts to be annotated are already split to sentences prior to your annotation. Each markable, i.e., the annotation you will generate in this stage, is called *SentenceOpinionAnalysisResult*. The following subsections explain the guidelines for the three attributes of this markable taking the annotation order into account.

### A.1.1 Guidelines for the *topic_relevant* attribute

*topic_relevant* attribute is the first attribute of the *SentenceOpinionAnalysisResult* markable and has four options to choose from: *not_set*; *none_given*; yes and no. *not_set* is the default selection prior to the annotation. It enables us and you to discover any unprocessed instances.

Some of the documents you will work with will have a given topic which was attributed by the source or the author such as "Apple iPod" or "Yahoo! finance service". If no topic is given for a document, please select the option *none_given* for the current sentence.

If a topic is given for the document from which the current sentence originates, you have to decide whether the current sentence deals with the given topic. This distinction is necessary because people sometimes drift off when writing. For example in a review for a certain BMW car people might compare it to a model by Chrysler and therefore list information or utter opinions about the Chrysler car. In this case, sentences which only deal with the Chrysler model are considered to be off-topic and not relevant for the purpose of our analysis. Therefore, in this case *topic_relevant* should be set to no.

So please only annotate sentences as *topic_relevant* = yes if:

1. the topic itself is discussed;
2. certain aspects / properties / features of the topic are discussed.

Otherwise annotate the sentences as *topic_relevant* = *no*. Example 1 illustrates two cases for the topic relevance:

(A.1) [**Topic: Eric Clapton**] I guess music should be judged based upon how much the soul is engaged or elevated. **(the sentence does not directly talk about Eric Clapton ⇒ *topic_relevant* = *no*)**

(A.2) Stevie Ray Vaughan and the undeservedly ignored Roy Buchanan, both RIP, are much better altogether. **(although a reader could infer that the two guitar players "Stevie Ray Vaughan" and "Roy Buchanan" are compared to Eric Clapton, there is no reference to the person Eric Clapton at all. Since you shall analyze each sentence as an independent unit you would set: ⇒ *topic_relevant* = *no*)**

Based on your decision for this attribute (if you selected *none_given* or yes), you will be presented the next attribute called *opinionated*, described in the next section.

### A.1.2 Guidelines for the *opinionated* attribute

The *opinionated* and *polar_fact* attributes both assess the evaluative character of the sentence regarding the topic.

***opinionated* attribute:** assesses whether the sentence contains one or more opinions about the given topic (if previously *topic_relevant* was set to yes) or whether the sentence contains any opinions (if previously *topic_relevant* was set to *none_given*).

Opinions are private states, i.e., utterances containing the ideas, beliefs, thoughts of a person towards an entity or regarding a situation. While expressing an opinion the writer assesses, judges or evaluates a subject matter, reflects his personal point of view about it. Therefore, contrary to facts, opinions are subjective, not falsifiable and not verifiable. They can vary from one person to another. The following criteria can help you in your decision:

- Does the sentence only report factual information?

- Is all information given in the sentence verifiable / falsifiable (for example by measuring something)?

If you answered both of these questions with "yes", the sentence is **not** opinionated and shall therefore be attributed with ***opinionated = no***.

- Does the sentence contain a personal evaluation of some kind?

If you answer this question with no, then the decision would be ***opinionated = no***. Example 2 illustrates sentences where the annotation is made according to the given criterion:

(A.3) I'm currently attending the educamp in Ilmenau.
**(verifiable fact ⇒ *opinionated = no*)**

(A.4) When I asked to speak to a supervisor they said they would have a manager contact me.
**(verifiable fact ⇒ *opinionated = no*)**

(A.5) Horrible experience with paypal.
**(personal evaluation of the experience ⇒ *opinionated* = yes)**

(A.6) Bought 6 guitars on ebay, purchased paypal's moneyback guarantee.
**(verifiable fact ⇒ *opinionated = no*)**

(A.7) Guitars were of poor quality so I tried to send back.
**(personal evaluation of the quality ⇒ *opinionated* = yes)**

(A.8) The movie was rather long.
**(personal evaluation of the duration of the movie ⇒ *opinionated* = yes)**

(A.9) The package was too big to send with US mail.
**(verifiable fact ⇒ *opinionated = no*)**

(A.10) The report is full of absurdities.
**(personal evaluation of the report ⇒ *opinionated* = yes)**

When deciding about the *opinionated* attribute, you should only consider the realis cases. In other words: the cases that happened or will happen. Please do not annotate sentences which describe a certain event, experience, state, ... which *might*, *could*, *can* or *possibly occur* if certain constraints are met. As a result, conditional or subjunctive sentences discussing hypothetical situations and their consequences for example as in "The iPod would be great if ... " should be annotated as *opinionated = no*. Watch out for such sentences! This aspect can be very subtle and is often only reflected by one word such as an "*if*". For example the sentence:

(A.11) eFax plus also seems like a great deal if you have a scanner and are without a fax machine.
**(there are two hints in this sentence which make it difficult: on the one hand the author says that** "eFax seems like a great deal" **which**

**reflects that this is not his opinion or the case and on the other hand there is a certain condition mentioned** "if you have a scanner and are without a fax machine" ⇒ ***opinionated* = *no***)

Please do not annotate sentences as opinionated or polar facts if the statements include certain conditions. Other examples for that are:

(A.12) Basically what I'm trying to say is that if you're travelling on your own and you get a good price this hotel is fine
**(certain circumstances are given under which the hotel is fine but this is not a universally valid statement ⇒ *opinionated* = *no***)

(A.13) I do have a fax machine, but since it is on the same line as my regular phone, it can be a pain to receive faxes.
**(The terms** "can be" **in the last phrase make the difference here. The author implies a certain condition, therefore, please do not annotate this as *opinionated* = *no***)

### A.1.3 Guidelines for the *polar_fact* attribute

Typically, explicit expressions of opinions communicate an attitude (also called valence or semantic orientation) which can be positive or negative about the topic being discussed. We will analyze this aspect of opinions in depth in the second stage. However, besides opinions, factual sentences can also result in a positive or negative impression of a given topic.

In reviews, writers occasionally explain their experiences with a certain product without explicitly uttering their opinions. Nevertheless, we can infer a positive or a negative evaluation regarding the topic. These sentences contain information which, if you read them and use your common sense or world knowledge, will give you a negative or positive impression. For such sentences we introduce the *polar_fact* attribute with the possible values of *not_set*; yes or no. If you encounter a *polar_fact*, i.e., *polar_fact* = yes, then you will be asked to mark the attitude towards the topic with the *polar_fact*_polarity attribute. Since the *polar_fact* annotation is done on the sentence level, it is also possible that both a positive and a negative fact are uttered in the same sentence , namely in thesame unit of annotation. *polar_fact*s can therefore have the *polar_fact*_polarity values: positive, negative and both. Example 3 illustrates example annotations for the *polar_fact* and *polar_fact*_polarity attributes:

(A.14) The computer crashed every day.
**(Crashing of a computer is typically an unwanted situation. From this sentence we can deduce a negative evaluation regarding the computer. Note that there is no explicit opinion being expressed, but the word crash has a negative connotation in most contexts ⇒ *polar_fact* = yes, *polar_fact*_polarity = negative)**

(A.15) The newly bought blender only worked for two hours.
**(The word** "only" **indicates that the blender did not meet the expectations of the writer, hence shifts the valence of the sentence**

**towards a more negative evaluation ⇒ *polar_fact* = *yes*; *polar_fact*_polarity = negative)**

(A.16) The team barely made it to the second round.
**(Similar to the previous sentence the lexical item barely causes a shift in the valence of the sentence indicating that the team did not meet the expectations ⇒ *polar_fact* = *yes*; *polar_fact*_polarity = negative)**

(A.17) I have used this email service for over a year now and it has never failed me.
**(The statement that something** "has never failed me" **describes a positive experience. Therefore ⇒ *polar_fact* = *yes*; *polar_fact*_polarity = positive)**

(A.18) On the one hand my UPS deliveries were always right on time, on the other hand the packages were sometimes highly damaged.
**(A positive and a negative factual aspect of UPS deliveries are presented. *polar_fact* = *yes*; *polar_fact*_polarity = both)**

The definitions about realis cases and subjunctives from Section A.1.2 also apply to the annotation of polar facts. Therefore please do not annotate sentences which imply a condition or solely describe a hypothetical state / event / ... as polar facts.

### Differentiation between opinions and polar facts

With the following examples we would like to emphasize the difference between opinions and polar facts:

(A.19) The double bed was so big that two large adults could easily sleep next to each other.
**(positive) Polar fact not an Opinion [Very little personal evaluation. We know that it's a good thing if two large adults can easily sleep next to each other in a double bed]**

(A.20) The bed was too small.
**(negative) Opinion not a Polar fact [No facts, just the personal perception of the bed size. We don't know whether the bed was just 1,5m long or the author is 2,30m tall]**

(A.21) The bed was blocking the door.
**(negative) Polar fact not an Opinion [Just a fact. We know that it's generally undesirable not to be able to open a door]**

(A.22) The bed was delightfully big.
**(positive) Opinion not a Polar fact [No fact just the personal perception of the bed size. We don't know whether the author was just 1,5m tall or the bed is 2,30m long]**

### A.1.4 Stage-1 annotation scheme and annotation steps

This section presents the annotation scheme for the first stage as a whole and how you should proceed with the annotation process. Table A.1 shows examples of the *SentenceOpinionAnalysisResult* annotations.

Table A.1: Example Annotations *SentenceOpinionAnalysisResult*

| **Attribute** | **Possible values** |
|---|---|
| *topic_relevant* | not_set (default) / none_given / yes / no |
| *opinionated*(appears only if *topic_relevant=none_given* or *topic_relevant*=yes) | not_set (default) / yes / no |
| *polar_fact* (appears only if *opinionated*=no) | not_set / yes / no |
| *polar_fact*_polarity (appears only if *polar_fact*=yes) | positive / negative / both |

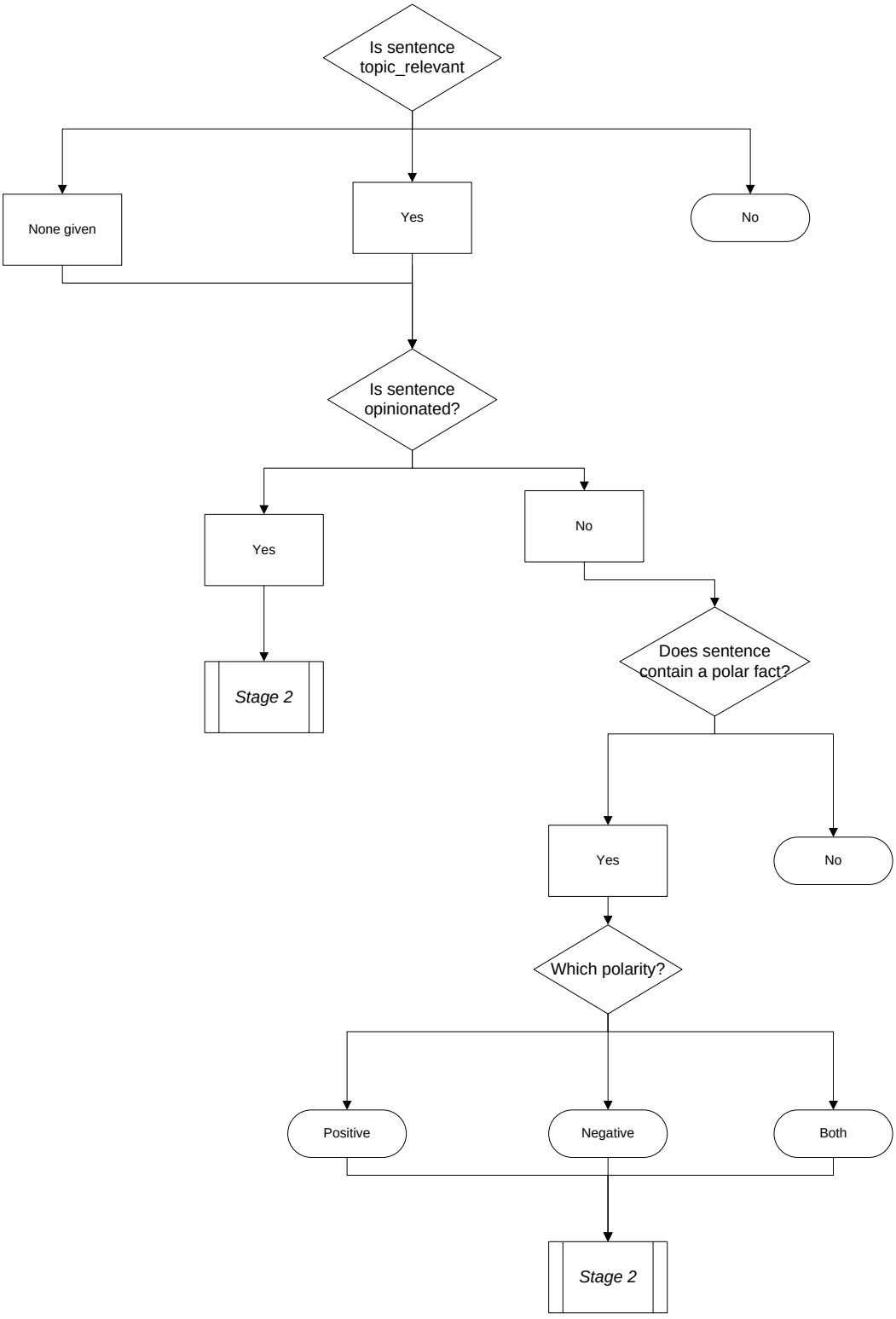
1. Read the sentence carefully. If no topic is provided with the document, mark *topic_relevant* as *none_given*.
2. If a topic is given, decide whether the sentence is relevant for the topic. If it is, mark *topic_relevant* as yes, otherwise mark *topic_relevant* as no.
3. If you marked *topic_relevant=none_given* or *topic_relevant*=yes, you will be presented with the *opinionated* attribute.
4. Think about whether the sentence contains a personal evaluation or a verifiable fact about the topic in a realis context (not a subjunctive sentence). If there is an opinion, i.e., personal evaluation, mark it as *opinionated = yes*, otherwise no.
5. If you marked *opinionated*=no, you will be presented with the *polar_fact* attribute. Think about whether you can clearly deduce a positive or negative impression about the topic of the sentence. If so, mark it as *polar_fact*=yes, otherwise no.
6. If you marked *polar_fact*=yes, then decide about the polarity of the evaluation(s) regarding the topic in the sentence.

The decision tree on the next page visualizes the process described above.

### A.1.5 Examples for the *SentenceOpinionAnalysisResult* markables

**[Topic: Hotel, hotel features]**

Figure A.1: Decision Tree For Sentence Level Annotation



(A.23) This is an older hotel, designed for the business traveller, undergoing remodelling.
**opinionated: NO, topic_relevant: YES, polar_fact: NO**
Reason: Only facts stated. The fact that the hotel undergoes remodeling does not imply a negative or positive connotation in this sentence / formulation - topic is the hotel

(A.24) Located in a more upscale, hotel area of Casablanca.
**opinionated: NO, topic_relevant: YES, polar_fact: NO**
Reason: Although this sentence qualifies as a polar_fact, there is no possible target (since it is omitted)! Therefore in such cases please refrain from annotating the sentence as opinionated or containing a polar fact - topic is the hotel

(A.25) I spent a quick night here as part of a tour.
**opinionated: NO, topic_relevant: NO, polar_fact: NO**
Reason: Only facts without positive or negative connotation stated - topic is the hotel

(A.26) The room was fine, of decent size and amenities.
**opinionated: YES, topic_relevant: YES: polar_fact: NO**
Reason: "fine" is a personal evaluation, the comment on the size is also subjective; therefore this is an opinion and not a polar_fact - topic is a room in the hotel

(A.27) The restaurant was terrific: the best chocolate ice cream ever and platters upon platters of breakfast choices.
**opinionated: YES, topic_relevant: YES: polar_fact: NO**
Reason: "terrific" gives a personal evaluation of the restaurant, same goes for the ice cream therefore this is an opinion and not a polar fact - topics are the restaurant of the hotel and a dish served in the restaurant of the hotel

**[Topic: Eric Clapton]**

(A.28) The most overrated guitar player in history.
**opinionated: YES, topic_relevant: YES, polar_fact: NO**
Reason: "overrated" is a notion of personal evaluation versus the general opinion - "guitar player" refers to the topic Eric Clapton

(A.29) He is certainly competent at playing the blues, but he doesn't impress me in the least.
**opinionated: YES, topic_relevant: YES, polar_fact: NO**
Reason: The evaluation of Eric Clapton's competence is a personal opinion, the same accounts for his ability to not impress the author - the sentence is about Eric Clapton's abilities and his impact on the author.

(A.30) I guess music should be judged based upon how much the soul is engaged or elevated.
**opinionated: NO, topic_relevant: NO, polar_fact: NO**

Reason: The authors just suggests how a certain aspect of people's notions should be, although this is a personal belief please generally refrain from annotating subjunctive sentences (the "should" gives you the hint here) - the people's perceptions of "soul" or the author's beliefs on that topic are no attribute directly related to Eric Clapton

(A.31) Stevie Ray Vaughan and the undeservedly ignored Roy Buchanan, both RIP, are much better altogether.
**opinionated: NO; topic_relevant: NO, polar_fact: NO**
Reason: Again, although an opinion is uttered, there is no target so please do not annotate this sentence as opinionated or factual - the only topics of the sentence are "Stevie Vaughan" and "Roy Buchanan" and not Eric Clapton

I guess music should be judged based upon how much the soul is engaged or elevated. This dude does neither for me.
**opinionated: NO; topic_relevant: yes; polar_fact: NO**
Reason: Although "This dude" refers to Eric Clapton, it is impossible to make a decision of the opinion or fact for this sentence as an individual unit since its content refers to certain aspects stated in a previous sentence - the topic is Eric Clapton

(A.32) There are those who believe Clapton to be a genius, but I'm not one of them.
**opinionated: NO; topic_relevant: YES, polar_fact: YES**
Reason: The author separates himself from a certain group of people which have a positive feeling about Eric Clapton and therefore gives the impressions that he has a rather negative attitude towards him. Therefore, the opinion is not explicitly stated but embedded in a fact - topic is Eric Clapton

(A.33) I don't hate the man, I just feel nothing, which is ultimately worse than full out hatred in my opinion.
**opinionated: NO; topic_relevant: YES, polar_fact: YES**
Reason: This is a difficult sentence, especially if you try to identify certain positive or negative elements in it. Due to the objective and analytical undertone it would be best to annotate it as not opinionated but a negative polar_fact - the topic is Eric Clapton

## A.2 Stage-2: Expression-level annotation process

In the second stage, we aim to analyze the *opinionated* and the *polar_fact* sentences in depth to gain more information about each individual opinion or evaluation and its target and holder. In this stage, you will be presented the sentences that are marked as *opinionated* or *polar_fact* by all the annotators in the previous stage. Sections A.2.1 and A.2.2 explain how to proceed with each type of sentence. Section A.2.3 shows the annotation scheme for the second stage as a whole and Section A.2.4 presents example annotations of polar facts and opinionated sentences.

### A.2.1 Processing *opinionated* sentences

An opinion can be analyzed as a combination of different attributes. Typically an opinion is uttered by some entity, towards another entity or situation where it has a certain connotation (a.k.a. attitude or semantic orientation or polarity) and intensity (strength) associated with it. Connotation is an implied value judgment or feelings associated with an opinion, i.e., whether the opinion holder intended to make a positive, negative or neutral evaluation with this opinion. From now on we refer to connotation as polarity in our study. We analyze opinions via four markable types:

1. **Holder:** is the entity who utters the opinion.
2. **Target:** is the entity or the situation which the opinion is about.
3. **Modifier:** is the lexical item(s) causing a shift in the strength of the opinion.
4. **OpinionExpression:** is the lexical item(s) instantiating the opinion, i.e., the expression from which we understand that there is a personal evaluation made.

All four markable types require marking a span of words and occasionally setting some attributes. Please note that the spans (=words) which you select are very important! This aspect greatly influences the quality of the annotated data. Therefore please choose wisely which term(s) you attribute to a certain markable, especially if you decide to include several terms into one markable. In the following we outline each type of markable in detail :

#### Holder markable

The holder markable represents the holder of an opinion in the sentence if it is other than the author himself. It has two attributes as *isReference* and *referent* which will be explained shortly.

Typically in product reviews people comment about their own experiences with a product, and therefore the opinion holder is the author most of the time. **In such cases where the author is the holder you should <u>not</u> create a holder markable.**

However, it can also be the case that the holder is some other entity as in "John Doe says that vacuum cleaners are useless." Here you should create a holder markable for the text span "John Doe". **While marking the text spans, you should always mark the minimum span of words fully describing the holder, you should not include articles or possessive pronouns, or a description of the holder in the span.** For instance, in "John Doe, the president of the broom factory, says that vacuum cleaners are useless." you should only mark "John Doe". Opinion holders are people most of the time, but they can also be organizations, governments, institutions etc.

If the opinion holder is other than the author and present in the sentence as a pronoun or another form of reference, you should set the *isReference* attribute to true otherwise set it to false. By default, this value is set to *not_set*. As it is the

case in the first stage,the *not_set* option is there to make sure that you do not skip stating your decision.

If the *isReference* is true, then you should resolve it to the referenced holder, i.e., look for the nearest antecedent of the reference in the previous sentences. The ideal procedure is the following:

1. Create the holder markable on the reference and set *isReference* to true
2. Find an antecedent of the reference
3. Create a holder markable on the antecedent (if there is none)
4. Return back to the original holder annotation (the one with the reference)
5. Create a link to the antecedent via the *referent* attribute.

Example 4 illustrates various holder annotations:

(A.34) New York Times food columnist Mark Bittman used to look down on the microwave for any sort of cooking beyond reheating leftovers or softening ice cream.
**(holder=Mark Bittman; *isReference*=false)**

(A.35) But after a couple of conversations with microwave cooking experts and a few experiments of his own, he said that the microwave is a more valuable tool in the kitchen than some of us give it credit for.
**(holder=he; *isReference*=true; *referent*=pointer to "Mark Bittman")**

(A.36) For any vegetable you would parboil or steam, the microwave works as well or better, and is faster.
**(no holder, not given explicitly)**

(A.37) A lot of interesting sessions, especially yovisto really impressed me.
**(no holder because the holder is the author)**

### Target markable

The target markable is used to annotate the target of the opinion in the sentence. Typically, the targets are nouns, but they can also be pronouns or complex phrases. The target markable is created analogous to the holder markable. In case if the target is a pronoun or another form of reference to a previously mentioned target, the *isReference* attribute should be set to true and the *referent* attribute should be set to point to the referenced target. You may set the referent pointer to the respective target even if it occurs in a different sentence.

Note that while annotating the targets please mark the minimum span describing the target, i.e., do not include any articles or unnecessary adjectives. **Include adjectives only if they constitute an integral part of the entity type specification. For instance, annotate "digital camera" vs. just "camera" or "external drive" vs. just "drive".** You can check whether a certain entity fulfills

this criterion by formulating a statement with "X is a type of Y". For example "a digital camera is a type of camera" or "an external drive is a type of drive", which in both cases makes sense and should therefore be annotated as targets. On the other hand, "a red house is a type of house" does not make sense. In the following, we present some example annotations of targets.

(A.38) New York Times food columnist Mark Bittman used to look down on the microwave for any sort of cooking beyond reheating leftovers or softening ice cream.
**(target=microwave; *isReference*=false)**

(A.39) But after a couple of conversations with microwave cooking experts and a few experiments of his own, he said that the microwave is a more valuable tool in the kitchen than some of us give it credit for.
**(target=microwave; *isReference*=false)**
**(target="tool"; *isReference*="true"; *referent*="microwave")**

(A.40) For any vegetable you would parboil or steam, the microwave works as well or better, and is faster.
**(target=microwave; *isReference*=false)**

(A.41) A lot of interesting sessions, especially yovisto really impressed me.
**(target=sessions; *isReference*=false)**
**(target=yovisto; *isReference*=false)**

(A.42) What is really cool is the automatic indexing of videos using ocr, and the possibility to tag a certain point in time and to post these tags in other social bookmarking systems.
**(target=automatic indexing of videos using ocr; *isReference*=false)**
**(target=possibility to tag a certain point in time; *isReference*=false)**
**(target=to post these tags in other social bookmarking systems; *isReference*=false)**

### Modifier markable

It is used to mark the words which cause shifts in the polarity of an opinion towards a target for example "not, very, hardly . . . ". They are the lexical items effecting the strength of an opinion. In other words, if you take them out, the opinion does not disappear, but the intensity of it will change.

The modifier markable is associated with the type attribute, which can be set to the values of *not_set* (the default value prior to annotation), negation, increase or decrease. Ultimately, the modifier annotation together with the OpinionExpression type annotation (described next) in a sentence constructs the whole opinion about the target. However, the OpinionExpression is the major actor. Example 6 illustrates example annotations for the modifier markable type:

(A.43) But after a couple of conversations with microwave cooking experts and a few experiments of his own, he said that the microwave is a more valuable tool in the kitchen than some of us give it credit for.

**(modifier=more; type=increase)**

(A.44) A lot of interesting sessions, especially yovisto really impressed me.
**(modifier=really; type=increase)**

(A.45) What is really cool is the automatic indexing of videos using ocr, and the possibility to tag a certain point in time and to post these tags in other social bookmarking systems.
**(modifier=really; type=increase)**

(A.46) His behavior during the presidential race was not very nice.
**(modifier=not; type=negation**
**modifier=very; type=increase)**

### OpinionExpression markable

It is used to mark the minimum span of words / phrases which actually instantiate an opinion. These words / phrases make the difference between the sentences which express an opinion and the sentences which express a fact. **While marking the span of the *OpinionExpression*, make sure not to include any modifiers - there is a separate markable for this purpose as mentioned above.**

The *OpinionExpression* markable has five attributes: *polarity, strength, modifier, holder* and *target* where three of them (*modifier, holder* and *target*) are pointers to the previously described markables.

**polarity and strength attributes:** *OpinionExpressions* invoke a negative or positive evaluation regarding the target. This is captured by the *polarity* attribute with the possible values of negative and positive. The intensity of the polarity is captured by the *strength* attribute with the possible values of weak, average and strong. This granularity is required since some terms have an inherently stronger impact than others. For example compare: "satisfying" (weak) - "good" (average) - "excellent" (strong) when used to evaluate the quality of a certain thing.

Some lexical items only reveal their polarity and strength when analyzed in their context e.g. "cool". Please set the polarity and strength attributes according to the context they occur in. **However, while doing so, do not take the effect of the modifier markable into account if there were any marked**. The term "disappointing" should be annotated with the polarity negative, even if the term "not" is preceding it.

For instance, in the sentence "The new generation strongly supports Barack Obama.", considering all the information given so far, we would mark "new generation" as the *holder*, "Barack Obama" as the *target*, "strongly" as the *modifier*. The *OpinionExpression* in this sentence is the word "supports". When deciding about its polarity and strength, we should look at it if the sentence were "The new generation supports Barack Obama." Therefore, the *polarity* should be set to positive and *strength* should be set to average, although from the original sentence you would infer a strong positive sentiment for "Barack Obama". In short, given a *modifier* and *OpinionExpression*, we will extract the overall evaluation of the opinion (modifier

+ OpinionExpression) based on your annotations. You just need to analyze each component as an independent unit.

Both the *polarity* and the *strength* attributes will be clearer with the examples presented below:

(A.47) New York Times food columnist Mark Bittman used to look down on the microwave for any sort of cooking beyond reheating leftovers or softening ice cream.
(OpinionExpression=look down; polarity=negative; strength=average ⇒ note that OpinionExpressions can originate from any word class.)

(A.48) But after a couple of conversations with microwave cooking experts and a few experiments of his own, he said that the microwave is a more valuable tool in the kitchen than some of us give it credit for.
(OpinionExpression=valuable; polarity=positive; strength=average ⇒ here, as you may have noticed, we judged the polarity of the opinion expression as if the word "more" was not there. "more" is annotated in a previous step as a modifier with "increase" type. The overall evaluation of this sentence will result from the merging of modifier and the opinion expression polarity after the annotation is complete.)

(A.49) A lot of interesting sessions, especially yovisto really impressed me.
(OpinionExpression=interesting; polarity=positive; strength=average) OpinionExpression=impressed; polarity=positive; strength=average there is also a modifier in the sentence, i.e., "really")

(A.50) What is really cool is the automatic indexing of videos using ocr, and the possibility to tag a certain point in time and to post these tags in other social bookmarking systems.
(OpinionExpression=cool; polarity=positive; strength=average again, in this sentence there is also the modifier "really")

(A.51) The bag was too big.
(OpinionExpression=too big; polarity=negative; strength=average ⇒ In this example we also include the word "too" within the OpinionExpression, since "The bag was big" alone would be a neutral statement. What creates the personal evaluation here is the "too", therefore it's part of the OpinionExpression.

(A.52) The Dell Latitude was a big disappointment.
(OpinionExpression=disappointment; polarity= negative; strength=average note that "big" would be annotated as a modifier)

The remaining three attributes describe the links between the *OpinionExpression* and the previously explained markable types:

*OpinionModifier* attribute: is a pointer to the modifier markable(s) in the current sentence which affect(s) the current OpinionExpression. Modifiers can be present, but do not have to. It's possible that several modifiers belong to one OpinionExpression!

*OpinionTarget* attribute: is a pointer to the target markable(s) which are commented on by the current OpinionExpression. It should always be possible to select the target(s) of the current OpinionExpression! One OpinionExpression can have more than one target.

*OpinionHolder* attribute: is a pointer to the holder who utters the current opinion. As we asked you to only annotate the holder if it's not the author, it's quite possible that no holder is present in a sentence.

### A.2.2 Processing *polar_fact* sentences

In the first stage of the annotation study we defined the *polar_fact* attribute, which was used to label statements which are not opinions, but invoke a positive or negative association in the reader. If a sentence is marked as a *polar_fact*, we are interested in the targets of such evaluations. When presented a polar fact in the second stage you need to create a *PolarTarget* markable analogous to the Target markable described in the previous section. However, a *PolarTarget* additionally has the polarity attribute with the possible values of positive and negative. Example 9 illustrates the target annotations for some *polar_facts*.

(A.53) The computer crashed every day.
**(target = computer; *isReference*=false; polarity=negative)**

(A.54) The newly bought blender only worked for two hours.
**(target = blender; *isReference*=false; polarity=negative)**

(A.55) The team barely made it to the second round.
**(target = team; *isReference*=false; polarity=negative)**

(A.56) The old car never failed me.
**(target = car; *isReference*=false; polarity=positive)**

Figure A.2 visualizes the decision tree for the *polar_fact* annotations.

### A.2.3 Stage-2 annotation scheme and annotation steps

Before we provide a sequence of annotation steps, we will give you an overview of the annotation scheme for the second stage as a whole in Table A.2.

**Expression level annotation steps:**

1. Read the sentence carefully. If the sentence is an opinionated sentence, proceed with step 2, else if it is a polar fact proceed with step 19.

2. **Holder:** Check whether the holder of the opinion is explicitly mentioned and if it is different from the author. If yes continue at 3, otherwise at 8.

3. Create the holder markable.

4. If the holder is a reference, set the "*isReference*" attribute to true, otherwise to false. If it is a reference, continue at 5, otherwise at 8.

Figure A.2: Decision Tree For *polar_fact* Annotations

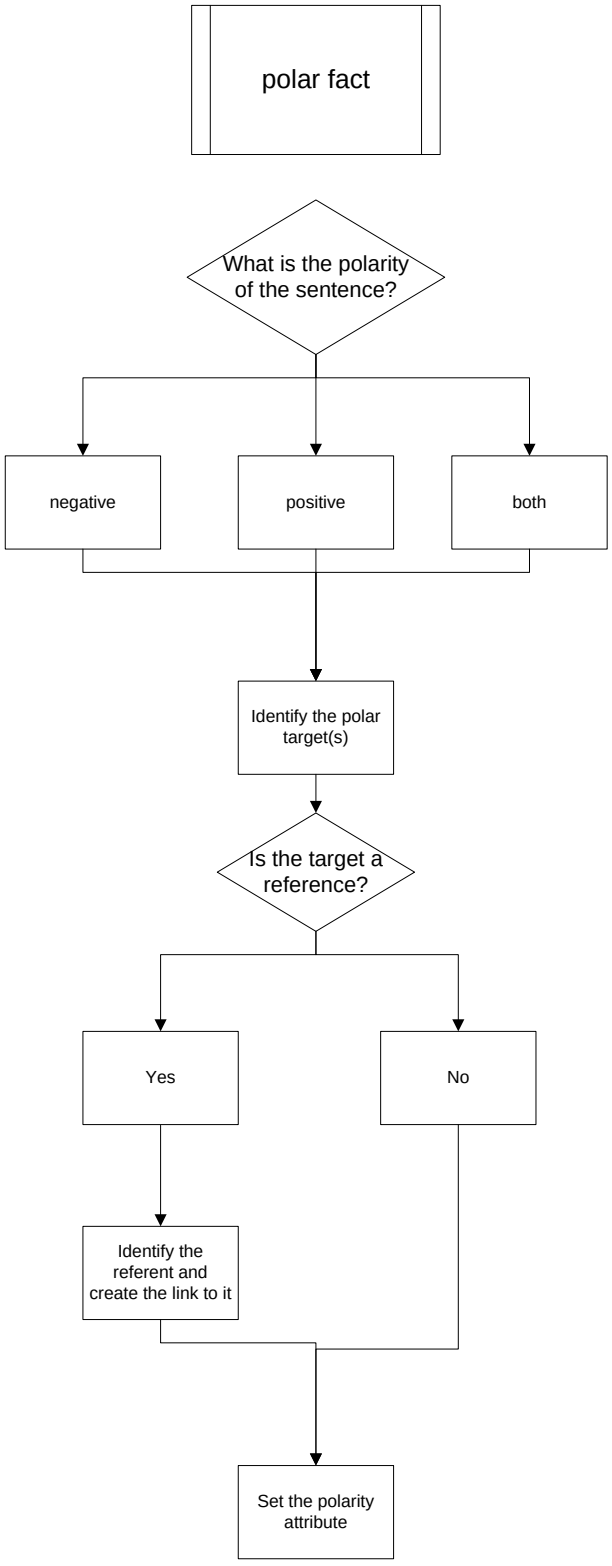


Table A.2: Stage-2: Expression-level markable types with their attributes and possible values

| **Markable** | **Attribute** | **Possible values** |
|---|---|---|
| Target | *isReference* | not_set / yes / no |
| | *referent* (set if *isReference*=true) | [pointer to target markable] |
| Holder | *isReference* | not_set / yes / no |
| | *referent* (set if *isReference*=true) | [pointer to holder markable] |
| Modifier | *type* | not_set / negation / increase / decrease |
| OpinionExpression | *strength* | not_set / average / weak / strong |
| | *polarity* | not_set / positive / negative |
| | *OpinionModifier* | [pointer(s) to modifier markable(s)] |
| | *OpinionTarget* | [pointer(s) to target(s)] |
| | *OpinonHolder* | [pointer(s) to holder(s)] |
| PolarTarget | *isReference* | not_set / yes / no |
| | referent (set if *isReference*=true) | [pointer to (Polar)Target markable] |
| | *polarity* | not_set / negative / positive |

5. Find the antecedent.
6. Check whether the antecedent was already marked as a holder in the referenced sentence. If there is no holder annotation on it in the referenced sentence, create one.
7. Create a pointer from the referring holder which has the *isReference*=true to its antecedent.
8. **Target:** Identify the target of the opinion, create the target markable.
9. If the target is a reference to another target, set the *isReference* attribute to "true", otherwise to "false". If it is a reference, continue at 10, otherwise at 13.
10. If the current target is a reference, locate the antecedent of the reference.
11. Check whether the antecedent was marked as a target in the referenced sentence. If there is no target annotation on it in the referenced sentence, create one.
12. Create a pointer to the resolved target from the referring target.

13. **Modifier:** Identify the word(s) which form the opinion.
14. Analyze the opinion. What causes the final semantic orientation of the opinion towards the target? Are there any modifiers such as negations which shift the polarity? Or are there any terms which shift the strength of the opinion? Annotate these modifiers if there are any and set their type accordingly.
15. **OpinionExpression:** Annotate the OpinionExpression. Mark the polarity and the strength of the expression - this should be done by analyzing the OpinionExpression as if no modifiers were present in the sentence. Determine the polarity and the strength of the OpinionExpression in the current context and set it accordingly.
16. If you identified any modifiers in 14, create a pointer to them.
17. Create a pointer to the target of the opinion which you identified in 8.
18. If you found any holder(s) in 2, create a pointer to the markable(s).
19. **PolarTarget:** Identify the target of the *polar_fact*. Annotate it analogous to the steps described 8-12.
20. Set the polarity intended for the target accordingly.

The complete decision tree for all elements of an opinion are shown in Figure A.3.

### A.2.4 Examples of stage-2 annotations

Table A.3 shows some example annotations. Please note that two or more OpinionExpressions can be created for the same holder or the target. What is crucial for an analysis is to capture each and every one of the OpinionExpressions separately. For instance, if you see a span like "... was excellent and terrific ... " for each evaluation, distinct markables "excellent" and "terrific" of the type OpinionExpression should be created. However, they may point to the same target or holder and even modifier. This way we aim to capture distinct opinions regarding an entity.

The number of markables created for each sentence is listed below the sentence itself. Note that for reasons of readability, we omitted the attributes of the target markable if they are no references.

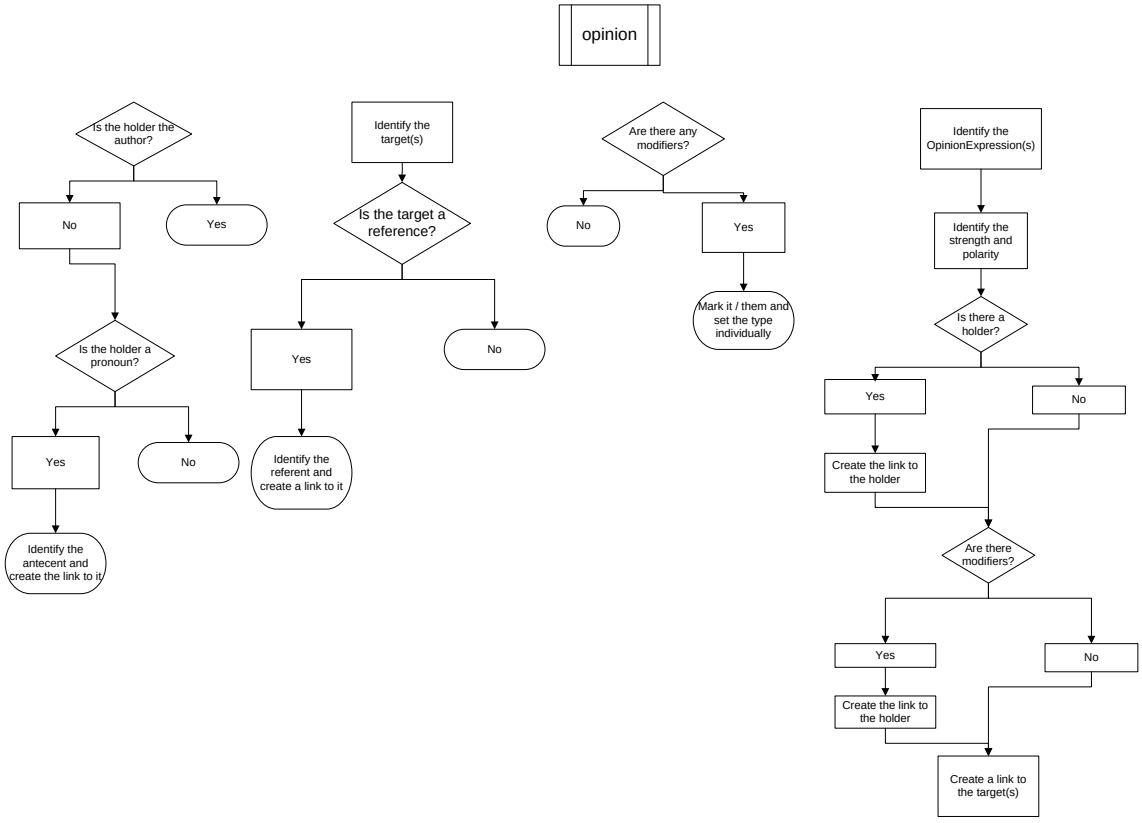
Table A.3: Stage-2: Expression-level markable types with their attributes and possible values

| **Sentence** | **Markables** |
|---|---|
| [**opinionated=yes**]<br><br><br><br><br><br>The room was fine, of decent size and amenities. (5 Markables) | **target**: room<br>**OpinionExpression**: fine<br>polarity: positive<br>strength: average<br>OpinionHolder: none<br>OpinionTarget: room<br>OpinionModifier: none |

| | |
|---|---|
| | **target**: size<br>**OpinionExpression**: decent<br>polarity: positive<br>strength: average<br>OpinionHolder: none<br>OpinionTarget: size<br>OpinionModifier: none |
| | **OpinionExpression**: amenities<br>polarity: positive<br>strength: average<br>OpinionHolder: none<br>OpinionTarget: room<br>OpinionModifier: none |
| [**opinionated=yes**]<br><br>The restaurant was terrific: the best chocolate ice cream ever and platters upon platters of breakfast choices. (4 Markables) | **target**: restaurant<br>**OpinionExpression**: terrific<br>polarity: positive<br>strength: strong<br>OpinionHolder: none<br>OpinionTarget: restaurant<br>OpinionModifier: none |
| | **target**: chocolate ice cream<br>**OpinionExpression**: best<br>polarity: positive<br>strength: strong<br>OpinionHolder: none<br>OpinionTarget: chocolate ice cream<br>OpinionModifier: none |
| [**opinionated=yes**]<br><br>New York Times food columnist Mark Bittman used to look down on the microwave for any sort of cooking beyond reheating leftovers or softening ice cream. (3 Markables) | **target**: microwave<br>holder: Mark Bittman<br>**OpinionExpression**: look down<br>polarity: negative<br>strength: average<br>OpinionHolder: Mark Bittman<br>OpinionTarget: microwave<br>OpinionModifier: none |
| [**opinionated=yes**]<br><br>But after a couple of conversations with microwave cooking experts and a few experiments of his own, it turns out that the microwave is a more valuable tool in the kitchen than some of us give it credit for. (4 Markables) | **target**: microwave<br>**target**: tool<br>referent: microwave<br>**modifier**: more<br>type: increase<br>**OpinionExpression**: valuable<br>polarity: positive<br>strength: average<br>OpinionHolder: none<br>OpinionTarget: tool |

| | |
|---|---|
| | OpinionModifier: more |
| [**opinionated=yes**]<br><br>For any vegetable you would parboil or steam, the microwave works as well or better, and is faster. (3 Markables) | **target**: microwave<br>**OpinionExpression**: well<br>polarity: positive<br>strength: average<br>OpinionHolder: none<br>OpinionTarget: microwave<br>OpinionModifier: none |
| | **OpinionExpression**: better<br>polarity: positive<br>strength: strong<br>OpinionHolder: none<br>OpionionTarget: microwave<br>OpinionModifier: none |
| [**opinionated=yes**]<br><br>The "we'll get to that" of eggplant was Bittman's biggest microwave revelation, calling his microwaved eggplant "mind-blowingly good." (4 Markables) | **target**: eggplant<br>**holder**: Mark Bittman<br>**modifier**: mind-blowingly<br>type: increase<br>**OpinionExpression**: good<br>polarity: positive<br>strength: average<br>OpinionHolder: Mark Bittman<br>OpinionTarget: eggplant<br>OpinionModifier: mind-blowingly |
| [**opinionated=yes**]<br><br>Aside from vegetables, the article also suggests puddings and crustless cakes can do wonders in the microwave. (3 Markables) | **target**: microwave<br>**OpinionExpression**: do wonders<br>polarity: positive<br>strength: strong<br>OpinionHolder: none<br>OpinionTarget: microwave<br>OpinionModifier: none |

Figure A.3: Decision Tree For Expression Level Annotations



# Appendix B

# $D_W$ Corpus Creation for LRT$_{wiki}$

The Wikipedia pages were retrieved and rendered using the Lynx text web-browser[1]. The rendered text website was then dumped and parsed as follows:

Parse the website dump sequentially while skipping lines which start with the following strings:

```
#[1]Edit this page
  (Redirected from
Jump to:
```

Stop the parsing process as soon as a line is encountered which contains one of the following strings (case insensitive):

```
edit] references
edit] external links
edit] related links
edit] see also
edit] notes
edit] further reading
You can help Wikipedia
Retrieved from "
Categories:
```

Or if a line is encountered which simply contains:

```
Views
```

Then remove all the numbered buttons which allow a user to edit a section with the following regular expression:

```
[[(d)+]edit]
```

Remove all hyperlink buttons which are rendered as numbers in square brackets:

```
[(d)+]
```

[1] http://lynx.isc.org/

And finally remove all Wikipedia "citation needed" buttons which match the following string:

```
[citation needed]
```